P.S.I.

PAR JACQUES DECONCHAT

P.S.I.

102 PROGRAMMES POUR HECTOR en BASIC 3X

Ouvrages d'initiation

— Visa pour l'informatique - Jean-Michel Jégo
— Visa pour le Basic - Jean-Michel Jégo
— Mon ordinateur - Jean-Claude Barbance
— Le logotron informatique - Jean-Pierre Petit
— L'ordinateur apprivoisé - François Picard et Danielle Shaw

La loi du 11 mars 1957 n'autorisant, aux termes des aliénas 2 et 3 de l'article 41, d'une part, que les «copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective», et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, «toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite» (aliéna 1er de l'article 40)

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 et suivants du Code Pénal

© Éditions du P.S.I. - B.P. 86 - 77402 Lagny cedex
1985
ISBN: 2-86595-237-1

JACQUES
DECONCHAT

102 PROGRAMMES POUR HECTOR en BASIC 3X

adapté par Jacques LABIDURIE

EDITIONS DU P.S.I.
1985

SOMMAIRE

Avant-propos 8
Organisation de l'ouvrage 10

Niveau 1

Instructions élémentaires 12
Le nombre mystérieux 18
Cube 20
Chaud, froid 22
Les allumettes 24
Allumettes à plusieurs 26
Calendrier perpétuel 28
Bataille navale 30
Golf 32
Solfège 34
Le jeu du 21 36
Pair - Impair 38
Devine un chiffre 40
Le jeu de Wytthof 42
Rebonds 44
Jackpot 46
Roulette 48
Couloirs 50
Echiquier 52
Le jeu de la fève 54
Jeu de l'oie aléatoire 56

Niveau 2

Dimensionnement d'un tableau - Recherche d'un élément dans un tableau et dans une chaîne - L'instruction DATA 60
Dé truqué 62
Visions 64
421 66
L'électricien fou 68
La course au 2Ø 70
Musique 72
Poker 74
Le jeu de saute-mouton 76
Lettres 78
Machinamot 80
Espion 82
Accrochez les wagons 84

Tirage d'un dé truqué 86
Tour de cartes 88
Le jeu de Marienbad 90
L'indésirable 92
Jour de la semaine 94
Le jeu de la vie 96

Niveau 3

Manipulation des chaînes 101
Un dé 102
Plus grand, plus petit 104
Ordre 106
Le pendu 108
Belote 110
Un taquin de calcul 112
Pendu inversé 114
Nicomaque 116
Course de chevaux 118
A travers 120
Mastermind 122
Idem 124
Horloge 1 126
Reines sur l'échiquier 128
Répétitions 130
Mots 132
Permutations 134
Bridge 136
Hanoï 138
Le loup et les agneaux 140

Niveau 4

Les techniques d'animation élémentaire 144
Tirage de dés 146
Jeu de loto 148
La bombe 150
Affichage publicitaire 152
Evasion 154
Chapeaux 156
Combat aérien 158
Le jeu du missile 160
Le singe 162
Le jeu du 15 164
Adresse 166
Sous-marin 168
Zombies 170

Diago 172
Décomaque 174
Chasse au sous-marin 176
Clavier 178
Le lézard d'Alice 180
Calculatrice 182

Niveau 5

Techniques spécifiques de HECTOR 187
Compose 188
Losanges 190
Mathieu 192
Ardoise magique 194
Dessin de dés 196
Smokey 198
Course de voitures 200
Echecs 202
Danger spatial 204
Tic Tac Toe 206
Horloge 2 208
Dames 210
Quatre à quatre 212
Anagrammes 214
Œufs 216
Taquin de lettres 218
Cibles 220
Gouttes 222
Obstacles 224
Captures dans l'espace 226
Anomalie 228
Plusieurs en un 230
Parcours du cavalier 232
Mur de briques 234
Othello 236

Avant-propos

Vous trouverez dans ce livre plus de 100 programmes (essentiellement des jeux), rédigés en Basic B3X, pour l'ordinateur individuel HECTOR HRX. Certains programmes seront d'ailleurs utilisables sans grande modification sur HECTOR 2 HRX en Basic III.

Les programmes proposés sont très courts, et l'ordre dans lequel ils sont présentés n'est bien évidemment pas arbitraire : il s'agit d'initier le lecteur, à travers la réalisation et l'utilisation de programmes de jeux de bonne qualité, à la connaissance et à la maîtrise progressive de cet outil extraordinaire qu'est un ordinateur individuel.

La brièveté des programmes proposés permettra de les taper facilement, sans risque d'erreur. Vous pourrez bien sûr les utiliser comme base pour la réalisation de jeux plus performants.

Les programmes sont abondamment commentés pour vous aider à comprendre leur fonctionnement, et les pages de présentation vous permettront de retrouver facilement les caractéristiques propres aux instructions Basic qu'ils utilisent.

Les programmes n'utilisent pas toutes les particularités d'un appareil comme le HECTOR, par ailleurs très bien détaillées dans le manuel de base.

Le niveau 5 donne une idée sur l'utilisation de certaines techniques un peu spécifiques, comme les instructions graphiques et sonores. Ces techniques pourront être reprises avec profit pour améliorer les programmes précédents.

Vous trouverez, à la fin de chaque texte de présentation, quelques suggestions pour des améliorations possibles, ou des modifications, qui pourraient rendre le jeu proposé plus attrayant, plus performant ou plus complet.

Tous les jeux de cet ouvrage ont été testés plusieurs fois et les listings sont une photocopie des sorties sur imprimante.

Une erreur de listing serait donc tout à fait exceptionnelle. Si, malgré tout, vous en relevez une, assurez-vous bien que vous avez correctement recopié le programme.

Voici quelques conseils qui pourront vous aider à déceler les "bogues" malencontreuses :

– Ne confondez pas le Ø (zéro) avec le O (lettre o).

– Comptez vos lignes de programme pour ne pas en oublier.

– Assurez-vous que le nom de vos variables reste le même d'un bout à l'autre du programme.

– N'oubliez pas les virgules et points-virgules nécessaires.

– Vérifiez que vos incrémentations sont placées au bon endroit...

Et surtout, armez-vous de patience !

Organisation de l'ouvrage

Les jeux sont classés par ordre de difficulté croissante, suivant 5 niveaux :

- **Le niveau 1** permet l'étude des instructions élémentaires du langage Basic : **LET** (optionnel ici), **PRINT, INPUT, IF...THEN, FOR...NEXT, GOTO, GOSUB, ON...GOTO, ON...GOSUB.**

- **Au niveau 2,** on utilisera les instructions de confection et de manipulation des éléments d'un tableau, et de recherche d'un élément dans une chaîne : **DIM, R(I), MID$, READ, DATA.**

- **Au niveau 3,** on utilise la panoplie complète des instructions de traitement des chaînes de caractères : **MID$, RIGHT$, LEFT$.**

- **Le niveau 4** fait appel à deux instructions très particulières qui facilitent la réalisation de jeux interactifs : **INKEY$** et **INSTR$.**

- Enfin, **le niveau 5** nous fait découvrir des instructions plus spécifiques : instructions graphiques, identification de couleurs sur l'écran, instruction sonore : **GR$, SCREEN (X, Y), PLOT, BOX, POINT (X, Y), TONE.**

Chaque jeu est présenté d'une façon très systématique, par une courte description, suivie de l'étude détaillée de chacune des lignes importantes du programme. Après quoi les indications nécessaires au bon fonctionnement du jeu sont données, ainsi que quelques éléments pouvant conduire à des extensions ou des améliorations du jeu proposé.

Le programme, fonctionnant sur le HECTOR, est ensuite reproduit, ainsi qu'un exemple d'exécution. Les listings ont été obtenus directement d'après les originaux de l'auteur, sur une imprimante matricielle, ce qui devrait garantir contre tout risque de disfonctionnement. Les minuscules ne sont pas utilisées. Avant de lancer un programme, assurez-vous que vous êtes en majuscules !

Pour vous arrêter en cours de programme, tapez CNT et C toujours suivi d'ENTREE.

La touche rouge de retour chariot (à droite du clavier) sera appelée "ENTREE" dans tout l'ouvrage.

Niveau 1

Nom du programme	*Thèmes et particularités*
Le nombre mystérieux	Jeu de déduction, pour un joueur. Plusieurs niveaux.
Cube	Jeu de dessin : perspective sur un écran.
Chaud, froid	Logique et hasard, pour un joueur. Plusieurs niveaux.
Les allumettes	Jeu de réflexion, contre l'ordinateur.
Allumettes à plusieurs	Jeu de réflexion et de stratégie, pour plusieurs joueurs.
Calendrier perpétuel	Confection d'un calendrier, mois par mois.
Bataille navale	Un seul navire à trouver, mais quelques particularités.
Golf	Un parcours de 18 trous, pour un seul joueur.
Solfège	Apprenez le solfège à l'aide de votre ordinateur.
Le jeu du 21	Jeu de hasard, l'ordinateur tient la banque.
Pair - Impair	Un jeu de casino en trois temps (1 ou plusieurs joueurs).
Devine un chiffre	Un jeu très facile, pour un ou plusieurs joueurs.
Le jeu de Wytthof	Un jeu de réflexion assez simple, pour deux joueurs.
Rebonds	Un jeu d'adresse très visuel, pour un joueur.
Jackpot	Jeu de hasard, pour un joueur.
Roulette	Une simulation élégante du jeu très connu.
Couloirs	Un jeu de logique, à découvrir.
Echiquier	Une base graphique pour vos propres créations.
Le jeu de la fève	Un jeu de réflexion, pour deux joueurs.
Jeu de l'oie aléatoire	Un très vieux jeu, revu et corrigé par l'ordinateur, deux joueurs.

Instructions élémentaires

Nous ne retiendrons ici que les instructions fondamentales, ou instructions de programmation ; beaucoup d'autres commandes, ou fonctions particulières, seront utilisées à l'occasion, par exemple :

– ***RND (I,J) pour obtenir un nombre aléatoire entre I et J.***

– ***INT pour prendre la partie entière d'un résultat.***

– ***REM ou ! pour placer un commentaire.***

– ***END pour terminer un programme.***
etc.

Les instructions fondamentales de la programmation, utilisées dès le niveau 1, et présentes dans pratiquement tous les programmes sont :

LET (optionnelle) ; **PRINT ; INPUT ; GOTO └─┘ ; IF ___ THEN └─┘ ; FOR ... NEXT ; GOSUB ... RETURN.**

soit seulement sept instructions de base (et deux instructions dérivées) que nous allons maintenant examiner plus en détail.

L'INSTRUCTION LET OU INSTRUCTION D'AFFECTATION

C'est l'instruction par excellence des langages de programmation évoluée, puisque son rôle essentiel est de permettre, pour le programmeur, une gestion transparente de la mémoire. Cela signifie, en pratique, que pour ranger une quantité en mémoire, par exemple 12, il n'est pas nécessaire de savoir dans quelle case de la mémoire on va la déposer : il suffit de lui affecter un nom autorisé, et c'est par ce nom que la quantité sera par la suite retrouvée et manipulée.

– **Si la quantité est numérique,** le nom sera une simple lettre de l'alphabet, ou un ensemble de plusieurs chiffres ou lettres, commençant obligatoirement par une lettre.

– **Si la quantité est alphanumérique**, la même règle s'applique pour le nom, qui devra cependant être suivi du symbole $, et la variable devra être écrite entre guillemets.

Exemple :
LET A1 = 12 ; la variable numérique A1 prend la valeur 12

LET NOM $ = "RENOIR" ; la variable alphanumérique NOM$ prend la valeur RENOIR.

Sur HECTOR, comme dans la plupart des Basic, le LET est devenu optionnel, aucune ambiguïté n'étant possible, on écrira donc : A = 12 ; NOM$ = "TOTO" ; etc.

L'INSTRUCTION PRINT OU INSTRUCTION DE SORTIE (ECRAN)

C'est une instruction de dialogue qui permet à la machine d'afficher sur l'écran un texte, le résultat d'un calcul ou une variable. C'est le seul moyen que possède l'ordinateur individuel pour faire connaître ses résultats. Ainsi, PRINT A1 provoque l'affichage de 12 (si l'on a fait auparavant A1 = 12), PRINT A$ provoque l'affichage de BONJOUR (si l'on a fait A$ = "BONJOUR"). Utilisé après un effacement d'écran, le PRINT provoque l'affichage sur la 1re ligne, en haut, à gauche.

Un certain nombre de paramètres peuvent être utilisés pour obtenir sur l'écran un affichage convenable :

- **PRINT TAB**(N) provoque sur la ligne d'impression courante un affichage commençant à la énième colonne à partir de la gauche ($\emptyset \leqslant N \leqslant 37$).

- Un ; (**point-virgule**) en fin de PRINT permet soit d'enchaîner une autre impression sans répéter l'instruction PRINT, soit d'indiquer que la prochaine impression sera, en l'absence d'indication contraire, effectuée exactement à la suite de l'impression précédente.

- La , (**virgule**) provoque une tabulation horizontale automatique prédéfinie.

Lors de la frappe, **l'instruction PRINT** pourra être abrégée en tapant **?** (**point d'interrogation**). Lors d'une impression, tout nombre est précédé d'un signe ; mais le signe positif n'est pas affiché, un espace précédera donc automatiquement l'impression de tout résultat positif.

L'INSTRUCTION INPUT OU INSTRUCTION D'ENTREE (CLAVIER)

Cette instruction, qui provoque un arrêt et une attente de l'appareil, ***permettra d'entrer un nombre, ou un texte, qui sera affecté à la variable suivant l'ordre INPUT.*** Ceci est fait en mode conversationnel, c'est-à-dire au fur et à mesure des besoins ; en effet, lorsque HECTOR rencontre une instruction INPUT, il s'arrête et vous rend la main ; on peut alors lui entrer une information de nature convenable (nombre ou texte), et lui signaler que cette entrée est terminée à l'aide de la touche **ENTREE,** ce qui lui rendra la main.

Si l'information entrée n'est pas de nature convenable (vous tapez, par exemple, un texte sur un INPUT X, qui attend un nombre), le message

REDO FROM START est affiché, et l'appareil se remet en attente.

Il est possible, comme pour le PRINT, d'afficher des messages avant l'entrée de la variable, et d'utiliser l'instruction CURSOR.

On pourra, par exemple, écrire :

INPUT "VOTRE NOM, S'IL VOUS PLAIT" ; A$

LES INSTRUCTIONS STANDARD, SPECIAL ET SCALE

HECTOR dispose de deux alphabets, dits STANDARD et SPECIAL.

– Le premier permet d'afficher 22 lignes de 38 caractères (numérotées à partir de Ø). Il est possible de travailler en mode calque avec COVER Ø (l'affichage d'un caractère n'efface pas celui qui s'y trouvait auparavant) ou en mode papier avec COVER 1.

– L'alphabet SPECIAL ne permet que l'affichage des majuscules, en indiquant une échelle avec l'instruction SCALE I, J. Avec SCALE 1,1 on dispose de 28 lignes de 28 caractères ; avec SCALE 1,2, on a 14 lignes de 28 caractères, etc.

L'INSTRUCTION CURSOR

HECTOR dispose d'une instruction de positionnement d'écran un peu particulière : CURSOR X, Y.

En SPECIAL, X et Y sont les "coordonnées caractères" (colonne, ligne d'affichage).

En STANDARD, X et Y sont les coordonnées en pixels (points élémentaires de l'écran graphique). Ainsi, pour se positionner en X, Y on tape ; CURSOR X*6+9, 226–Y*1Ø.

L'INSTRUCTION GOTO OU INSTRUCTION DE RENVOI INCONDITIONNEL

Une des particularités du langage Basic, qui le rend facilement accessible, est que ***chaque ligne de programme doit être précédée d'un numéro,*** ces lignes étant ensuite exécutées par l'ordinateur individuel en séquences, dans le sens des numéros croissants. Pour diverses raisons, il pourra être nécessaire au cours du déroulement d'un programme, de ne pas suivre cet ordre séquentiel très rigide ; on utilisera alors ***l'instruction GOTO, qui permet de détourner le fonctionnement du programme de son cours normal*** quitte d'ailleurs à le reprendre ensuite par un autre GOTO. Le renvoi doit, par ailleurs, impérativement se faire sur un numéro de ligne existant dans le programme (GOTO 2Ø, par exemple).

Pour donner un peu de souplesse à cette instruction de renvoi, il existe une instruction très puissante, qui permet de renvoyer à n'importe quel nu-

méro de ligne, selon la valeur d'une variable numérique : **c'est l'instruction ON└─┘GOTO└─┘** ; on écrira par exemple :

ON X GOTO 1ØØ, 2ØØ, 3ØØ et le renvoi sera fait à la ligne 1ØØ si X vaut 1, à la ligne 2ØØ si X vaut 2, et 3ØØ si X vaut 3.

L'INSTRUCTION IF└─┘THEN└─┘OU INSTRUCTION DE TEST

On peut l'utiliser de deux façons : d'abord, en rupture de séquence conditionnelle, sous la forme :

IF condition = vraie THEN numéro de ligne.

C'est l'usage le plus courant de ce test, qui permet de renvoyer à une autre partie de programme, si par exemple X = Ø, ou si A = B, etc. Les **opérateurs logiques NOT, OR** ou **AND** peuvent d'ailleurs être utilisés dans l'écriture de la condition.

On pourra aussi utiliser le test sous la forme :

IF condition THEN instruction ;

l'instruction présentée à la suite de THEN peut être une affectation A = 12, une impression PRINT A, une entrée INPUT B $, ou même un autre test IF └─┘THEN └─┘.

On peut par exemple écrire :

IF A > 3 THEN IF B < 7 THEN PRINT A + B

L'INSTRUCTION FOR...NEXT OU INSTRUCTION DE BOUCLES

On est très souvent conduit, en programmation, à vouloir exécuter un nombre entier de fois une certaine suite d'instructions (par exemple, faire avancer un mobile de la case 6 à la case 2Ø, etc.). Plutôt que d'utiliser un test et un renvoi, il sera préférable de faire appel à une ***boucle,*** la variable de boucle prenant ses valeurs dans les entiers ou les décimaux, et le ***pas*** de cette variable (c'est-à-dire la quantité dont elle augmente ou diminue à chaque passage) pouvant aussi prendre ses valeurs dans les entiers ou les décimaux.

Ainsi, si l'on écrit :

```
FOR I =  )  Ø TO 2Ø STEP 2
         }  Instructions
NEXT I   )
```

Les instructions contenues dans la boucle seront exécutées pour I = Ø, pour I = 2, ... et ceci jusqu'à I = 2Ø, soit 11 fois. La variable I peut être

utilisée pour faire un calcul dans la boucle, mais ne devra pas être modifiée dans la boucle. On peut, bien sûr, sortir de la boucle, avant I = 2Ø, par un test portant une quantité quelconque.

Si ***le pas de la boucle (STEP)*** n'est pas précisé, il sera automatiquement pris égal à 1.

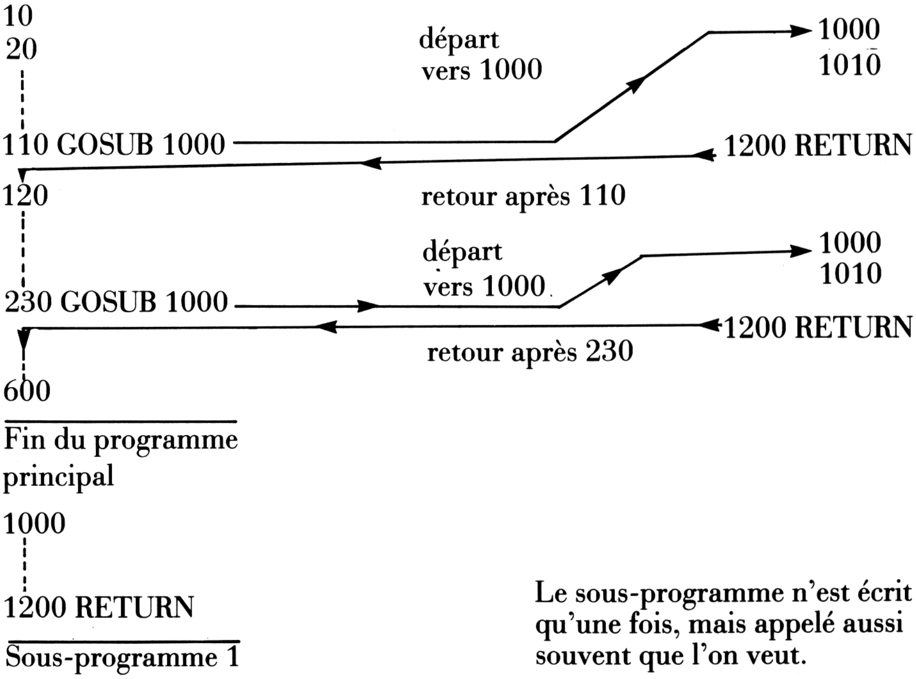
Il est possible d'emboîter plusieurs boucles de noms différents, et d'omettre l'indication de la variable dans le NEXT.

L'INSTRUCTION GOSUB OU INSTRUCTION D'APPEL DE SOUS-PROGRAMME

Lorsqu'une même suite d'instructions doit être exécutée plusieurs fois, et appelée à partir de plusieurs endroits du programme, on évitera de la réécrire plusieurs fois. On la rédige une fois pour toutes, et on la place en dehors du programme proprement dit (en lui donnant, par exemple, un numéro de ligne supérieur au numéro de la dernière ligne du programme). Indépendamment des difficultés que l'on pourra rencontrer au niveau du nom des variables, lorsque l'on sera désireux d'utiliser cette séquence, il suffira de l'appeler par **l'instruction GOSUB numéro de ligne.** Le programme ira dès lors "faire un détour" dans la partie du programme appelé, mais il ne pourra revenir et poursuivre son déroulement normal que si il rencontre dans le programme appelé une instruction **RETURN,** ou instruction de retour.

Voici le schéma théorique de fonctionnement :

Programme principal :



Un sous-programme pourra lui-même éventuellement appeler un autre sous-programme, et ainsi de suite.

On peut aussi appeler un sous-programme de façon conditionnelle, en utilisant l'instruction ON␣ GOSUB ␣.

Par exemple ON X GOSUB 1000, 2000, 3000 permettra d'exécuter le sous-programme placé en 1000 si X = 1, en 2000 si X = 2, en 3000 si X = 3.

Le nombre mystérieux

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de déduction pour un joueur.
Difficulté du jeu : facile.

LE JEU :
On demande à l'ordinateur de fabriquer un nombre de n chiffres (avec n compris entre 1 et 7).
On devra ensuite essayer de le découvrir, avec le moins d'essais possible, en recevant les seuls renseignements TROP GRAND ou TROP PETIT.

LE PROGRAMME :
La ligne 5Ø permet de fabriquer le nombre à deviner.
Les entrées sont faites par les lignes 7Ø à 8Ø.
La ligne 9Ø teste si le résultat est exact.
La ligne 1ØØ signale un essai trop petit, et la ligne 11Ø un essai trop grand.
Le compteur d'essais est en ligne 12Ø.
Les lignes 2ØØ à 23Ø s'occupent de l'affichage du résultat, et de la gestion de la partie suivante.

POUR JOUER :
On lancera le programme par RUN.
On obtient un premier affichage NOMBRE DE CHIFFRES auquel on répond par un entier entre 1 et 7, suivi de ENTREE.
On obtient alors l'affichage ESSAI N° 1, suivi de la question VOTRE NOMBRE.
On entre un nombre, et l'on presse ENTREE.
On obtient alors une réponse TROP PETIT, ou TROP GRAND, ou encore, si l'on a trouvé juste un affichage de VOUS AVEZ TROUVE... EN... ESSAIS. Si le résultat n'est pas exact, on revient à ESSAI N° 2, et VOTRE NOMBRE.
En fin de partie, on obtient la question AUTRE PARTIE (O/N) ; on répond O pour faire une autre partie (suivi de ENTREE).
On peut arrêter à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
On pourra imaginer de faire deviner au joueur autre chose qu'un chiffre (par exemple, un mot, en utilisant l'ordre alphabétique, ou encore autre chose...).

```
10 'LE NOMBRE MYSTERIEUX
20 CLS
30 INPUT"NOMBRE DE CHIFFRES ... :";N
40 IF N>7 THEN GOTO 20
50 X=INT(RND(0,10^N-1))
60 J=1
70 PRINT:PRINT"-- ESSAI N̄.";J
80 INPUT"   VOTRE NOMBRE ... :";M
90 IF M=X THEN GOTO 200
100 IF M<X THEN PRINT"== TROP PETIT"
110 IF M>X THEN PRINT"== TROP GRAND"
120 J=J+1
130 PRINT:PRINT
140 GOTO 70
200 CLS
210 PRINT"VOUS AVEZ TROUVE";X;"EN";J;"ESSAIS."

220 PRINT:INPUT"AUTRE PARTIE (O/N) :";R$
230 IF R$="O" THEN GOTO 20
```

Cube

REFERENCES :
Difficulté de programmation : aucune.
Catégorie de jeu : dessin.
Difficulté du jeu : aucune.

LE JEU :
Le programme proposé se contente de réaliser à l'aide d'instructions PRINT une vue d'un cube en perspective cavalière.
Il pourrait être par exemple incorporé à un jeu basé sur les capacités de reconnaissance des formes géométriques élémentaires.

LE PROGRAMME :
Le dessin du cube est réalisé par l'impression de points, en utilisant deux boucles : la boucle I (3Ø-12Ø), et la boucle I (13Ø-18Ø).
Le nom de l'objet représenté est affiché en ligne 19Ø (ici : CUBE).
Le positionnement du curseur est réalisé par le sous-programme en 1ØØØ-1Ø1Ø.

POUR JOUER :
Il suffit de lancer l'exécution par RUN, et de regarder.

EXTENSIONS POSSIBLES :
La haute résolution permettra d'avoir un tracé plus précis.
On pourra imaginer un jeu utilisant des parallélépipèdes rectangles de différentes tailles, avec affichage par exemple des 3 côtés, et il faudrait trouver leur volume, etc.

```
10 'CUBE
20 CLS
30 FOR I=0 TO 9
40 C=7+I:L=0:GOSUB1000:PRINT"."
50 C=I:L=7:GOSUB1000:PRINT"."
60 C=7+I:L=10:GOSUB1000:PRINT"."
70 C=I:L=17:GOSUB1000:PRINT"."
80 C=0:L=7+I:GOSUB1000:PRINT"."
90 C=10:GOSUB1000:PRINT"."
100 C=7:L=I:GOSUB1000:PRINT"."
110 C=17:GOSUB1000:PRINT"."
120 NEXT I
130 FOR I=0 TO 7
140 C=I:L=7-I:GOSUB1000:PRINT"."
150 C=10+I:GOSUB1000:PRINT"."
160 L=17-I:GOSUB1000:PRINT"."
170 C=I:GOSUB1000:PRINT"."
180 NEXT I
190 C=25:L=18:GOSUB1000:PRINT"CUBE"
999 END
1000 CURSORC*6+9,226-L*10
1010 RETURN
```

Chaud, froid

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

LE JEU :
Un des plus vieux jeux du monde, toujours aussi pratiqué, mais cette fois, c'est l'ordinateur qui vous sert de partenaire : il va en effet cacher un objet à un certain emplacement d'une grille dont vous aurez vous-même choisi les dimensions. Pour le trouver, il ne vous donnera que quelques indications sous la forme TU BRULES, ou C'EST FROID, etc.
Vous devez réussir à découvrir l'objet caché, avec le moins d'essais possible (votre score sera affiché en fin de partie).

LE PROGRAMME :
Les lignes 4Ø et 5Ø définissent la position de l'objet.
La ligne 9Ø calcule la distance par rapport aux deux coordonnées que vous avez entrées aux lignes 6Ø-7Ø.
La ligne 1ØØ regarde si vous avez gagné.
Les lignes 12Ø-13Ø normalisent la distance (qui doit avoir un sens, quelles que soient les dimensions de la grille initiale).
La ligne 14Ø effectue le départ, s'il y a lieu, vers l'un des sous-programmes d'affichage.

POUR JOUER :
On lance le programme par l'ordre RUN, puis on introduit les dimensions de la grille (carrée) et l'on frappe ENTREE.
Dès l'affichage suivant (DONNEZ X), on entre la coordonnée X supposée, suivie de ENTREE.
On obtient ensuite l'affichage DONNEZ Y, et l'on entre Y suivi de ENTREE.
On obtient en réponse un texte, qui est variable selon que l'on sera plus ou moins loin de l'objet cherché (cela va de COMPLETEMENT GELE à BRULANT). Lorsque l'on a trouvé, on obtient l'affichage du nombre d'essais effectués.
Pour sortir du programme avant la fin de la partie, il suffit de taper CNT et C.

EXTENSIONS POSSIBLES :
Si l'on dispose de la couleur, il sera intéressant de trouver un moyen plus visuel pour symboliser la distance, par exemple des pavés allant du rouge au bleu, aux emplacements essayés.
On peut aussi travailler dans un espace à trois dimensions, cacher plusieurs objets, etc.

```
10 'CHAUD,FROID
20 SEED TIME(0)
25 CLS
30 INPUT"DIMENSION DE LA GRILLE :";D
40 X=INT(RND(1,D))
50 Y=INT(RND(1,D))
55 CLS
60 PRINT:INPUT"DONNEZ X :";A
70 PRINT:INPUT"DONNEZ Y :";B
80 S=S+1
90 T=ABS(X-A)+ABS(Y-B)
100 IF T=0 THEN GOTO 6000
110 PRINT"VOUS ETES ";
120 N=INT(SQR(D)/T)+1
130 IF N>5 THEN N=5
140 IF T<=SQR(D) THEN ON N GOTO 1000,2000,3000
,4000,5000
150 PRINT"COMPLETEMENT GELE"
160 GOTO 60
1000 PRINT"A PEINE TIEDE."
1010 GOTO 60
2000 PRINT"TIEDE."
2010 GOTO 60
3000 PRINT"CHAUD."
3010 GOTO 60
4000 PRINT"TRES CHAUD."
4010 GOTO 60
5000 PRINT"BRULANT."
5010 GOTO 60
6000 CLS
6010 PRINT"VOUS AVEZ TROUVE";X;"/";Y;"EN";S;"C
OUPS"
```

Les allumettes

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : contre HECTOR.
Difficulté du jeu : moyenne.

LE JEU :
Le joueur devra en début de partie choisir une certaine quantité d'allumettes, ainsi que la prise maximum autorisée. Il décide ensuite s'il commence ou s'il laisse commencer HECTOR. Chacun prend à son tour autant d'allumettes qu'il le désire, dans la limite du nombre maximum autorisé. Celui qui prend la dernière allumette est considéré comme perdant.

LE PROGRAMME :
Le jeu du joueur commence en ligne 7Ø.
Celui de HECTOR commence en ligne 11Ø.
La ligne 8Ø contrôle la légalité du coup du joueur.
La ligne 9Ø gère le changement de joueur.
Le coup de HECTOR est calculé dans une boucle non formelle, de la ligne 12Ø à la ligne 15Ø.
Le résultat est affiché par la ligne 1Ø2Ø.
Si la partie est terminée, la ligne 1Ø3Ø le détecte et envoie en ligne 2ØØØ, pour le message de fin.

POUR JOUER :
On commence l'exécution par RUN, et, à l'affichage du message NOMBRE TOTAL, on entre le nombre d'allumettes, suivi de ENTREE. Puis on entre la prise maximum autorisée, suivie de ENTREE. Si le joueur veut commencer, il frappe ensuite Ø ; sinon il frappe 1, puis ENTREE.
Après chaque coup, le nombre d'allumettes restantes est affiché, et HECTOR affiche VOUS RETIREZ lorsque c'est au joueur de prendre.
Celui qui prend la dernière allumette a perdu.

EXTENSIONS POSSIBLES :
On pourra bien sûr envisager un affichage graphique couleur plus spectaculaire, imaginer de laisser le hasard choisir le nombre d'allumettes et la prise maximum, etc.

```
 10 'LES ALLUMETTES
 20 CLS
 30 INPUT"NOMBRE TOTAL .... :";N
 40 INPUT"PRISE MAXIMUM ... :";I
 50 INPUT"QUI COMMENCE (VOUS:0/MOI:1) :";D
 60 IF D=1 THEN GOTO 110
 70 PRINT:INPUT"VOUS RETIREZ :";P
 80 IF P<>INT(P) OR P<1 OR P>I THEN GOTO 70
 90 D=1
 100 GOTO 1010
 110 S=I+D
 120 T=(N-S)/(I+D)
 130 IF T=INT(T) THEN GOTO 200
 140 S=S-D
 150 GOTO 120
 200 P=ABS(S-D)
 210 IF P=0 THEN P=D
 220 D=0
 1000 PRINT:PRINT"JE RETIRE :";P
 1010 N=N-P
 1020 PRINT"IL RESTE";N
 1030 IF N=0 THEN GOTO 2000
 1040 GOTO 60
 2000 PRINT:PRINT
 2010 IF D=1 THEN PRINT"=== J'AI GAGNE ...":ELS
E PRINT"BRAVO. VOUS AVEZ GAGNE..."
```

Allumettes à plusieurs

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : logique (plusieurs joueurs).
Difficulté du jeu : assez difficile.

LE JEU :
Dans ce jeu pour plusieurs joueurs, le meneur de jeu devra, en début de partie, choisir un nombre d'allumettes, et indiquer à HECTOR le nombre de joueurs.
A chaque coup, le joueur qui a le trait est autorisé à retirer du tas d'allumettes restantes jusqu'au double de ce que vient de retirer le précédent joueur ; celui qui retire la dernière allumette a gagné (ou perdu, si l'on modifie légèrement les règles).

LE PROGRAMME :
Les lignes 10 à 40 permettent les différentes entrées.
Les lignes 70 à 100 réalisent les impressions.
La ligne 120 contrôle le coup de chaque joueur (par rapport au coup précédent).
La ligne 180 gère les tours (remise à 1 du numéro du joueur).
La ligne 160 réalise le test de fin, et la ligne 210 affiche le numéro du joueur gagnant (avec la règle prévue initialement).

POUR JOUER :
Après avoir lancé le programme par RUN, on entre le nombre d'allumettes initial, suivi de ENTREE, puis le nombre de joueurs (et ENTREE).
Le premier joueur ne pourra enlever qu'une allumette, le deuxième pourra en enlever 1 ou 2 ; le suivant 1, 2 ou 1, 2, 3 ou 4 selon le coup précédent, et ainsi de suite ; le retrait maximum autorisé est à chaque fois affiché par HECTOR.
Lorsque la dernière allumette a été prise, on obtient l'affichage LE JOUEUR... A GAGNE, et la partie est terminée.

EXTENSIONS POSSIBLES :
On peut convenir que celui qui prend la dernière allumette est éliminé du jeu, et recommencer la partie entre les joueurs restants qui seront ainsi éliminés les uns après les autres.
On peut aussi imaginer une représentation graphique des tas d'allumettes, tirer au hasard le nombre à enlever en début de partie, etc.

```
10 'ALLUMETTES A PLUSIEURS
20 CLS
30 INPUT"NOMBRE TOTAL ........ :";N
40 INPUT"NOMBRE DE JOUEURS ... :";J
50 F=1
60 K=1
70 CLS
80 PRINT"TOUR DU JOUEUR N.";K
90 PRINT"TOTAL RESTANT :";N
100 PRINT"MAXIMUM A ENLEVER :";F
110 INPUT"-- VOTRE COUP :";C
120 IF C=0 OR C>F THEN GOTO 110
130 N=N-C
140 IF N<0 THEN N=N+C:GOTO 110
150 F=C+C
160 IF N=0 THEN GOTO 200
170 K=K+1
180 IF K>J THEN K=1
190 GOTO 70
200 CLS
210 PRINT"LE JOUEUR N.";K;"A GAGNE."
```

Calendrier perpétuel

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : gadget publicitaire.
Difficulté du jeu : aucune.

LE JEU :
Ce programme vous permettra de confectionner tous les ans votre calendrier ; vous pourrez ainsi vous en servir pour envoyer à vos amis un calendrier du mois de leur naissance, ou pour leur souhaiter une bonne fête, etc.

LE PROGRAMME :
Les lignes 3Ø et 4Ø vous permettront d'entrer le mois et l'année.
Les lignes 5Ø, 6Ø et 7Ø calculent le numéro du premier jour du mois.
La ligne 8Ø calcule le nombre de jours du mois (il n'est pas tenu compte ici des 29 février, pour simplifier).
La boucle A (12Ø-15Ø) réalise l'impression du calendrier proprement dit.

POUR JOUER :
Après avoir lancé le programme par l'instruction RUN, on devra introduire l'année, sous la forme, par exemple 1985, suivie de ENTREE.
On obtient alors la question MOIS, à laquelle il convient de répondre par le numéro du mois, suivant l'ordre usuel (1 pour JANVIER, 2 pour FEVRIER, etc.).
On fait ensuite ENTREE, et le calendrier est alors affiché sur l'écran.

EXTENSIONS POSSIBLES :
On pourra sans difficulté prévoir l'affichage des 29 février, prévoir une option permettant l'affichage pour une année complète, prévoir l'introduction du nom des mois en clair, etc.

```
10 'CALENDRIER PERPETUEL
20 CLS
30 INPUT"POUR QUELLE ANNEE .... :";A
40 INPUT"POUR QUEL MOIS ....... :";M
50 D=INT(365.25*(A+(M<3)))+1
60 D=D+INT(30.6*(M+1-(M<3)*12))-INT((INT((A+(M
<3))/100)-7)*.75)
70 D=D-7*INT(D/7)
80 M=31-(-(M=4) OR -(M=6) OR -(M=9) OR -(M=11)
)+(M=2)*3
90 PRINT
100 PRINT"DIM LUN MAR MER JEU VEN SAM"
110 PRINT
120 FOR A=1 TO M
130 PRINTTAB(4*(D+A-1)+1);A;
140 IF D+A-1=6 THEN PRINT:D=D-7
150 NEXT A
160 PRINT
```

Bataille navale

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexion.
Difficulté du jeu : facile.

LE JEU :
Une bataille navale a priori assez simple, avec un seul navire caché au hasard dans une grille de 1Ø cases sur 1Ø cases.
A chaque coup, on devra tirer en donnant les coordonnées H (horizontale) et V (verticale) de la case visée, et HECTOR répond en affichant la distance séparant le navire invisible de la case visée. Mais, si l'on tire trop près, le navire risque de se recacher au hasard.

LE PROGRAMME :
La boucle I (4Ø-7Ø) dessine le terrain de jeu.
Les lignes 9Ø-1ØØ définissent l'emplacement du navire.
Les coordonnées du tir sont entrées aux lignes 11Ø et 12Ø, et la ligne 13Ø affiche le coup sur l'écran. La distance est calculée en ligne 14Ø, et la ligne 15Ø regarde si l'on a ou non fait mouche.
La ligne 17Ø intervient pour déplacer le navire si les coups sont rapprochés.
La ligne 18Ø permet l'affichage de la distance.

POUR JOUER :
Le terrain de jeu est affiché après avoir frappé RUN. A l'affichage de LIGNE, on répond en donnant le numéro de la ligne horizontale sur laquelle on désire tirer. On tape ensuite ENTREE et, à la question COLONNE, on répond en donnant le numéro de la colonne verticale suivi de ENTREE.
Si on a tiré trop près, on obtient l'affichage PFFFF... qui signifie que le navire ennemi a changé d'emplacement.
Si on a tiré sur la bonne case, on obtient l'affichage du nombre total de tirs effectués pour gagner.

EXTENSIONS POSSIBLES :
Elles sont innombrables, puisqu'aussi bien la bataille navale est l'un des jeux les plus répandus qui soit. On peut introduire deux joueurs, jouer contre HECTOR, mettre plusieurs navires, des sous-marins, etc.

```
10 'BATAILLE NAVALE
15 SEEDTIME(0)
20 COVER 1:CLS
30 PRINT"    0  1  2  3  4  5  6  7  8"
40 FORI=0TO8
50 PRINTI;" .  .  .  .  .  .  .  .  ."
60 PRINT
70 NEXT I
80 S=1
90 X=INT(RND(0,8))
100 Y=INT(RND(0,8))
110 C=0:L=19:GOSUB1000:INPUT"LIGNE :";H
120 C=19:L=19:GOSUB1000:INPUT"COLONNE :";V
130 C=3*V+3:L=2*H+1:GOSUB1000:PRINTS
140 D=INT(SQR((X-H)*(X-H)+(Y-V)*(Y-V)))
150 IF D=0 THEN 200
160 S=S+1
170 IF D=1 THEN C=0:L=20:GOSUB1000:PRINT"PFFFF
...";:GOTO 90
180 C=0:L=20:GOSUB1000:PRINT"DISTANCE  :";D;"
    "
190 GOTO 110
200 CLS
210 PRINT"TROUVE EN";S;"COUPS."
1000 CURSORC*6+9,226-L*10
1010 RETURN
```

Golf

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez facile.

LE JEU :
Il s'agit d'essayer de réaliser, avec le moins de coups possible, un parcours de golf de 18 trous. Pour cela, il vous faut frapper la balle avec une force plus ou moins grande, jusqu'à obtenir sa chute dans le trou.
Mais attention, il peut y avoir un certain aléa sur chaque coup, et vous serez amené souvent à revenir en arrière (il faudra alors faire précéder l'indication de la force par un signe −).

LE PROGRAMME :
La ligne 7Ø permet de définir la position du trou, et la ligne 8Ø dessine le terrain de jeu.
Le déplacement à réaliser est calculé en ligne 12Ø.
La boucle I (14Ø-19Ø) réalise le déplacement de la balle.
La ligne 2ØØ regarde si le coup est, ou non, gagnant ; s'il l'est, on recommence (ligne 25Ø) jusqu'à ce que l'on ait réalisé 18 trous (boucle S (4Ø-25Ø) ; fin de boucle en 27Ø, avec affichage de fin).

POUR JOUER :
On obtient l'affichage du terrain en frappant RUN ainsi que le dessin de la balle sur sa position de départ, et le texte : TROU 1. FORCE ; on doit alors répondre en donnant la force désirée (un nombre entre 1 et 3ØØ...).
Dès que l'on aura tapé ENTREE, la balle recommence à se déplacer ; elle peut alors soit s'arrêter avant le trou (auquel cas on recommence), soit tomber dedans (on passe alors au trou suivant), soit le dépasser : on devra alors frapper avec une force négative.
Le jeu se termine avec l'affichage 18 TROUS EN ... COUPS.

EXTENSIONS POSSIBLES :
Il sera possible d'étendre le jeu pour faire participer plusieurs joueurs, d'améliorer la qualité graphique, d'utiliser une représentation spatiale du terrain (se référer à certains jeux d'arcade), d'avoir un terrain plus accidenté, des obstacles, de changer de club...

```
 10 'GOLF
 20 SEEDTIME(0)
 30 SPECIAL
 40 FOR S=1 TO 18
 50 CLS
 60 X=1
 70 T=INT(RND(9,25))
 80 CURSOR0,7:FOR I=0 TO 27:PRINT"=";:NEXT
 90 CURSORT,7:PRINT" ":CURSORX,6:PRINT"O"
 100 CURSOR15,10:PRINT"                    "
 110 CURSOR0,10:PRINT"TROU ";S;:INPUT "FORCE ";
R
 120 F=INT(R*.16-(ABS(R)*(SQR(RND(0,1))+.1)^2)/
1000)
 130 IF F=0 THEN GOTO 100
 135 SOUND2,22
 140 FOR I=SGN(F) TO F STEP SGN(F)
 150 X=X+SGN(F)
 160 IF X<27 AND X>=1 THEN CURSOR X-SGN(F),6:PR
INT" ":CURSOR X,6:PRINT"O"
 170 IF X>=26 THEN X=26
 180 IF X<=1 THEN X=1
 190 NEXT I
 195 HUSH
 200 IF X=T THEN GOTO 230
 210 E=E+1
 220 GOTO 80
 230 CURSOR X,6:PRINT" ":CURSOR X,7:PRINT"O"
 235 LASER
 240 PAUSE 1
 250 NEXT S
 260 CURSOR 0,14
 270 PRINT"18 TROUS EN";E;"COUPS."
 280 STANDARD
```

Solfège

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : facile.

LE JEU :
Ce programme vous propose une initiation élémentaire au solfège, en vous permettant d'apprendre à reconnaître facilement les différentes notes, qui apparaissent les unes après les autres sur une portée ; après l'affichage de la note, vous devez entrer son nom en clair et votre score sera affiché en permanence.
Le programme est initialement prévu pour la clé de sol, mais il sera très facilement adaptable à toute autre clé.

LE PROGRAMME :
La boucle I (5Ø-7Ø) dessine une portée.
La note est choisie en ligne 8Ø et affichée en lignes 9Ø-1ØØ-11Ø.
La ligne 12Ø affiche le score.
La ligne 13Ø vous interroge.
La ligne 14Ø s'occupe de la fin de la partie.
Les lignes 15Ø et 16Ø effectuent le décodage de la note entrée.
La ligne 18Ø contrôle le résultat.

POUR JOUER :
Après avoir lancé le programme par RUN, la portée sera affichée, ainsi que la première note.
Vous répondrez à la question NOTE en donnant clairement le nom de la note (do, ré, mi, fa, sol, la ou si), suivi de ENTREE.
Le score vous indiquera si c'est exact ou non, et une nouvelle note sera affichée.
Pour arrêter, il suffira de répondre Ø suivi de ENTREE, à la question NOTE.

EXTENSIONS POSSIBLES :
On pourra tout d'abord facilement modifier le programme pour lui permettre de vous faire travailler les différentes clés. On pourra aussi améliorer le graphisme, introduire la couleur, le son (lorsque l'on a répondu juste, on entendra la note, par exemple). On pourra aussi réaliser suivant le même principe un jeu vous permettant de reconnaître des accords, etc.

```
 10 'SOLFEGE
 30 S=0:P=0
 40 CLS:SPECIAL:PEN2
 50 FOR I=1 TO 5
 60 CURSOR 0,2*I+1:PRINT"=====================
======"
 70 NEXT I
 80 X=INT(RND(0,11))
 85 PEN1:BEEP
 90 CURSOR13,X:PRINT"!"
 100 CURSOR13,X+1:PRINT"!"
 110 CURSOR12,X+2:PRINT"O"
 120 CURSORO,22:PRINT"SCORE :";S;"SUR";P;"ESSAI
S."
 130 PRINT:INPUT" QUELLE EST LA NOTE : ";R$
 140 IF R$="O" THEN PEN2:STANDARD:END
 150 Y=(1 AND R$="FA") + (2 AND R$="MI") + (3 A
ND R$="RE") + (4 AND R$="DO")
 160 Y=Y+(5 AND R$="SI") + (6 AND R$="LA")
 170 P=P+1
 180 IF Y=X OR Y=X-7 THEN S=S+1
 190 GOTO 40
```

Le jeu du 21

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu contre HECTOR.
Difficulté du jeu : facile.

LE JEU :
On devra entrer une mise au début du jeu, et on pourra lancer les dés en tapant 1 ; à chaque lancer, on obtiendra un nouveau total, qui augmentera peu à peu, et se rapprochera de 21. On devra essayer de se rapprocher le plus possible de 21, sans dépasser ce nombre. Lorsque l'on décidera de s'arrêter, il suffira de taper Ø pour laisser jouer HECTOR qui essaiera alors d'avoir un meilleur score.
A la fin de chaque partie, le gain sera affiché ainsi que le total des gains (ou des pertes).

LE PROGRAMME :
Les lignes 8Ø et 9Ø réalisent le contrôle de celui qui doit jouer (joueur ou HECTOR).
Le sous-programme (1ØØØ-1Ø1Ø) lance les dés.
Les lignes 11Ø-12Ø calculent le total fait par le joueur et l'affichent.
Le total de HECTOR est calculé en ligne 19Ø et affiché par la ligne 2ØØ.
La ligne 22Ø contrôle la fin du jeu pour HECTOR.
Les lignes 26Ø-27Ø affichent les résultats, et la ligne 3ØØ cherche quel est le gagnant.
Le gain (ou la perte) sera affiché par la ligne 32Ø, et le total par la ligne 35Ø.
La ligne 38Ø contrôle la fin de partie.

POUR JOUER :
Taper RUN pour lancer le jeu.
A la question VOTRE MISE, entrer la mise initiale, suivie de ENTREE, et taper 1 pour lancer le dé. Le premier tirage est affiché sous la forme VOUS AVEZ... et on devra refaire 1 suivi de ENTREE, tant que l'on désire recommencer à tirer.
Dès que l'on a terminé, on tape Ø et ENTREE pour laisser jouer HECTOR (qui tient ici le rôle de la banque).
Le gain (ou la perte, s'il y a le signe −) sera affiché en fin de jeu, ainsi que le total des gains (ou pertes) réalisé depuis le début. Si l'on désire arrêter de jouer, on devra taper N en réponse à la question ON CONTINUE (O/N).

EXTENSIONS POSSIBLES :
Ce jeu sera évidemment plus intéressant s'il est pratiqué à plusieurs, et il sera aisé de modifier le programme en conséquence. On pourra aussi dessiner le dé, ou mieux, utiliser des cartes, ce qui permettra alors de faire un jeu de Black Jack.

```
10 'JEU DU 21
20 SEEDTIME(0)
30 G=0
50 CLS
60 T=0
70 INPUT"VOTRE MISE :";M
80 INPUT"TAPER 1 POUR TIRER, 0 POUR FINIR :";R

90 IF R=0 THEN GOTO 150
100 GOSUB 1000
110 T=T+X
120 PRINT"===  VOTRE TOTAL EST DE ";T
130 GOTO 80
150 CLS
160 PRINT:PRINT"===  LA BANQUE JOUE ..."
170 S=0
180 GOSUB 1000
190 S=S+X
200 PRINT:PRINT"     TOTAL DE LA BANQUE :";S
210 PAUSE 1
220 IF S>=T OR S>17 THEN GOTO 250
230 GOTO 180
250 PRINT
260 PRINT"VOUS AVEZ .... :";T
270 PRINT
280 PRINT"LA BANQUE A .. :";S
290 PRINT
300 IF S>21 OR (S<T AND T<=21) THEN GOTO 320
310 M=-M
320 PRINT:PRINT"GAIN ... : $";M
330 PRINT
340 G=G+M
350 PRINT"VOTRE TOTAL ACTUEL : $";G
360 PRINT
370 INPUT"---  ON CONTINUE ? (O/N) : ";R$
380 IF R$="N" THEN END
390 GOTO 50
400
1000 X=INT(RND(1,6))
1010 RETURN
```

Pair-impair

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

LE JEU :
C'est un jeu de hasard, pour un ou plusieurs joueurs. HECTOR lance simultanément deux dés, non visibles et pose trois questions : quelle est la parité du total, quel est le total, et enfin donner la valeur de l'un des dés. On peut ensuite soit poursuivre le jeu, soit l'interrompre à tout moment.

LE PROGRAMME :
Les dés sont tirés par un sous-programme placé aux lignes 1ØØØ et 1Ø1Ø.
La ligne 8Ø tient le compte du nombre de parties, et les lignes 18Ø, 26Ø et 29Ø comptabilisent les points marqués.
La parité du total est déterminée par les lignes 13Ø et 14Ø (la ligne 14Ø compare avec la réponse du joueur).
La ligne 4ØØ et les suivantes réalisent l'affichage de fin de partie : total, nombre de points, nombre de parties jouées.
La ligne 44Ø provoque l'arrêt si l'on répond N à la question ON CONTINUE.

POUR JOUER :
On lance le programme par l'instruction RUN.
Il suffit ensuite de répondre aux diverses questions posées, en suivant scrupuleusement les indications données et en tapant ENTREE après chaque réponse.
Lorsque l'on a terminé une partie, on obtient l'affichage du total actuel et du nombre de parties sous la forme de, par exemple, 6 SUR 1Ø.
On obtient alors l'affichage de la question ON CONTINUE (O/N) et l'on devra répondre N suivi de ENTREE pour arrêter.

EXTENSIONS POSSIBLES :
Si l'on désire jouer à plusieurs, il sera intéressant de prévoir la possibilité de miser, et d'ajouter une gestion du score réalisé par chacun des joueurs. On pourra aussi prévoir une amélioration sur le plan de la présentation (utilisation de la couleur, etc.).

```
10 'PAIR-IMPAIR
20 SEEDTIME(0)
30 T=0
40 R=0
50 GOSUB1000
60 A=X
70 GOSUB1000
80 R=R+6
90 S=A+X
100 CLS
110 INPUT"TOTAL PAIR(0) OU IMPAIR(1) :";N
120 PRINT
130 P=S/2-INT(S/2)
140 IF N=P OR N-.5=P THEN GOTO 170
150 BEEP:PRINT"===  FAUX"
160 GOTO 190
170 BEEP:PRINT"===  EXACT"
180 T=T+1
190 PRINT
200 INPUT"TOTAL PRESUME :";N
210 IF N=S THEN GOTO 250
220 BEEP:PRINT"===  PERDU..."
230 PAUSE 1
240 GOTO 400
250 PRINT
260 T=T+2
270 BEEP:INPUT"BRAVO. DONNEZ UN DE :";N
280 IF N<>A AND N<>S-A THEN GOTO 220
290 T=T+3
300 PRINT
310 LASER:PRINT"BRAVO !":GOTO 230
400 CLS
410 PRINT"VOUS AVEZ UN TOTAL DE";T;"SUR";R
420 PRINT
430 INPUT"ON CONTINUE ? (O/N) : ";R$
440 IF R$="N" THEN END
450 GOTO 50
1000 X=INT(RND(1,7))
1010 RETURN
```

Devine un chiffre

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

LE JEU :
C'est un jeu de hasard, de conception très simple, mais néanmoins assez passionnant : le joueur introduit sa mise, et parie sur un chiffre, de 1 à 6 ; trois dés sont alors lancés ; si le chiffre joué apparaît deux fois, il double sa mise ; s'il apparaît trois fois, il triple sa mise.

LE PROGRAMME :
Le tirage des dés est réalisé par un sous-programme placé en lignes 1ØØØ et 1Ø1Ø.
L'affichage est réalisé par les lignes 15Ø et 16Ø.
Les lignes 18Ø et 21Ø calculent le gain dans les diverses configurations possibles.
La totalisation est effectuée en ligne 23Ø pour les gains, et en ligne 51Ø pour les pertes.
La variable R$, testée en ligne 55Ø, détermine la poursuite éventuelle du jeu.

POUR JOUER :
Le lancement du programme par RUN provoque un premier affichage VOTRE MISE ; on répond par un nombre suivi de ENTREE. On obtient alors un deuxième affichage DONNEZ UN CHIFFRE DE 1 A 6 ; on entre un chiffre suivi de ENTREE ; les 3 dés, repérés par les lettres A, B, C s'affichent alors, ainsi qu'un message indiquant votre gain ou votre perte. Puis une indication sur votre total actuel apparaît, ainsi que la question ON CONTINUE (O/N) ; taper O pour continuer, N pour arrêter (suivi de ENTREE).

EXTENSIONS POSSIBLES :
On pourra facilement introduire des contrôles (mise minimum, coup proposé de 1 à 6, etc.). On pourra aussi dessiner des dés avec un graphisme plus élaboré, introduire la possibilité de jouer à plusieurs, faire même participer HECTOR.

```
 10 'DEVINE UN CHIFFRE
 20 SEEDTIME(0)
 30 STANDARD
 40 G=0
 50 CLS
 60 INPUT"===  VOTRE MISE : $";N
 70 PRINT
 80 INPUT"DONNEZ UN CHIFFRE DE 1 A 6 :";D
 90 GOSUB1000
 100 A=X
 110 GOSUB1000
 120 B=X
 130 GOSUB1000
 140 C=X
 150 PRINT:PRINT:PRINT:PRINT"      A
B             C"
 160 PEN1:PRINT:PRINT:PRINT"     ";A;"
";B;"           ";C:PEN2
 170 T=0
 180 IF D=A THEN T=T+N
 190 IF D=B THEN T=T+N
 200 IF D=C THEN T=T+N
 210 IF T=0 THEN GOTO 500
 220 PRINT:PRINT:PRINT"===  VOUS GAGNEZ $";T
 230 G=G+T
 240 GOTO 520
 500 PRINT:PRINT:PRINT"===  PERDU ..."
 510 G=G-N
 520 PRINT:PRINT"===  TOTAL ACTUEL : $";G
 530 PRINT
 540 INPUT"ON CONTINUE ? (O/N) : ";R$
 550 IF R$="N" THEN END
 560 GOTO 50
 1000 X=INT(RND(1,7))
 1010 RETURN
```

Le jeu de Wytthof

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (deux joueurs).
Difficulté du jeu : moyenne.

LE JEU :
On dispose d'une grille carrée de 8 cases sur 8 cases. Un point apparaît, en dehors de la diagonale et les joueurs devront le déplacer à chaque tour, suivant seulement trois directions : droite, bas et diagonale, bas et droite. Le premier joueur qui arrive en (7,7) sera considéré comme gagnant, ou perdant, selon la convention faite au début.

LE PROGRAMME :
Les lignes 4Ø, 5Ø et 6Ø permettent de définir une position de départ, en dehors de la diagonale.
La ligne 8Ø et la boucle I (9Ø-11Ø) dessinent le terrain de jeu.
La ligne 13Ø imprime la position du point mobile.
Le test de fin est effectué en ligne 14Ø.
Le changement de joueur est assuré par la ligne 18Ø.
Les directions de déplacement disponibles, 1, 2 et 3, sont décodées par trois sous-programmes qui commencent aux lignes 1ØØØ, 2ØØØ et 3ØØØ.

POUR JOUER :
Taper RUN pour lancer l'exécution du programme. Le terrain de jeu s'affiche, ainsi que la position de départ.
Puis on obtient l'affichage JOUEUR 1 et en dessous la question DIRECTION (1, 2, 3), à laquelle on répond en tapant l'un de ces trois nombres, suivi de ENTREE.
C'est ensuite au tour du joueur 2.
Le jeu se termine lorsque l'un des joueurs arrive en case (7, 7).

EXTENSIONS POSSIBLES :
On peut tout d'abord améliorer les contrôles (la sortie du terrain n'est pas interdite, par exemple), améliorer ensuite l'affichage (graphisme, couleur), étendre le jeu à plusieurs joueurs avec un terrain agrandi, introduire un monstre qui va essayer de capturer un des joueurs, prévoir un programme qui permettrait à HECTOR de jouer, etc.

```
 10 'LE JEU DE WYTTHOF
 30 CLS:SPECIAL:SCALE1,1
 40 X=3*INT(RND(0,5))+5
 50 Y=INT(RND(0,5))
 60 IF 3*Y+5=X THEN GOTO 50
 70 PEN2:CLS
 80 PRINT"      0  1  2  3  4  5  6  7"
 90 FOR I=0 TO 7
 100 PRINT:PRINTI;"  .  .  .  .  .  .  .  ."
 110 NEXT I
 120 H=1:PEN1
 130 BEEP:CURSOR X,2*Y+2:PRINT"+"
 140 IF X=26 AND Y=7 THEN STANDARD:PEN2:END
 150 CURSOR 0,25:PRINT"JOUEUR";H;:INPUT"DIRECTI
ON 1,2,3";D$
 160 D=VAL(D$)
 170 ON D GOSUB 1000,2000,3000
 180 H=2-ABS(H-1)
 190 GOTO 130
 1000 X=X+3
 1010 RETURN
 2000 GOSUB1000
 3000 Y=Y+1
 3010 RETURN
```

Rebonds

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu d'adresse.
Difficulté du jeu : moyenne.

LE JEU :
Il utilise l'idée de la simulation d'un billard à bandes.
De part et d'autre de l'écran se trouvent la balle et le but visé : une série de chiffres et de lettres, en haut de l'écran, permettent de choisir l'endroit où l'on désire rebondir.
La balle va alors partir vers le haut de l'écran (vers le bas si l'on introduit un signe moins), y rebondir, revenir éventuellement frapper le bas, rebondir, et ainsi de suite jusqu'à atteindre la ligne de but ; si l'on est suffisamment près de la cible on a gagné et le nombre de rebonds sera alors affiché.

LE PROGRAMME :
La ligne 5Ø imprime les chiffres et les lettres en haut de l'écran.
Les lignes 6Ø et 7Ø définissent les positions initiales de la balle et de la cible.
Les lignes 8Ø et 9Ø effectuent leur impression.
Le symbole correspondant à la direction choisie est entré en ligne 1ØØ.
La boucle J (13Ø-17Ø) réalise le déplacement de la balle.
La ligne 16Ø imprime la nouvelle position.
Le changement de direction, lors de la rencontre d'un bord, est effectué par la ligne 15Ø, par renvoi dans une partie du programme indépendante, située aux lignes 22Ø à 24Ø.
La ligne 18Ø effectue l'impression de fin de partie.
Le nombre de rebonds est compté en ligne 23Ø.

POUR JOUER :
On lance le programme par RUN.
Le terrain de jeu est affiché, ainsi que les positions initiales de la balle et de la cible. On choisit l'endroit où l'on désire faire rebondir la balle (de 1 à Z), et on l'entre suivi de ENTREE.
La trajectoire est alors affichée, suivie s'il y a lieu de la mention BRAVO ; X REBONDS.
On peut lancer une autre partie en tapant une touche quelconque, autre que N, et s'arrêter à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
Il sera très intéressant, en utilisant un graphisme plus fin, de dessiner un véritable billard, avec son tapis vert, et d'indiquer la direction par un segment, symbolisant la queue de billard, se déplaçant autour de la bille.
On peut aussi ajouter des trous, mettre plusieurs billes, sonoriser les rebonds ou les heurts avec d'autres balles, les comptabiliser pour faire une partie à plusieurs joueurs, etc.

```
 10 'REBONDS
 30 CLS:SPECIAL:SCALE1,1
 40 CLS:PEN2
 50 PRINT"  ABCDEFGHIJKLMNOPQRSTUVWXYZ"
 60 X=INT(RND(2,20))
 70 Y=INT(RND(2,20))
 80 CURSOR0,X:PEN1:PRINT"#"
 90 CURSOR27,Y:PEN3:PRINT"+"
 100 PEN2:CURSOR0,24:INPUT"VOTRE COUP :  ";R$
 110 S=0:PEN1
 120 T=X
 130 FORJ=1TO27
 140 T=T-X/(ASC(R$)-64)
 150 IF T>=22 OR T<=0 THEN GOTO 220
 160 BEEP:CURSOR J,T:PRINT"#"
 170 NEXT J
 180 IF ABS (T-Y)<3 THEN LASER:PEN 2:PRINT"== B
RAVO :";S;"REBONDS"
 190 PRINT:INPUT"AUTRE PARTIE ? (O/N) :  ";R$
 200 IF R$="N" THEN PEN2:STANDARD:END
 210 RUN
 220 X=-X
 230 S=S+1
 240 GOTO 170
```

Jackpot

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : hasard (un joueur).
Difficulté du jeu : facile.

LE JEU :
C'est une variante simplifiée d'un jeu de hasard classique, dans lequel le joueur doit faire une mise. S'il obtient certaines combinaisons de symboles, il gagne (plus ou moins, selon sa mise et la combinaison obtenue).
Ici, les symboles sont remplacés par des chiffres, et le fait d'avoir trois chiffres identiques permet de gagner. Il existe de plus une combinaison particulière qui permet de remporter le Jackpot.

LE PROGRAMME :
La boucle I (6Ø-8Ø) permet le dessin de la machine, la ligne 9Ø affichant les emplacements (vides) des chiffres.
La mise sera entrée en ligne 1ØØ, et testée en ligne 11Ø : si on mise Ø, la partie est terminée (départ vers la ligne 2ØØØ).
Les lignes 12Ø, 14Ø et 16Ø permettent le tirage des 3 chiffres, en appelant un sous-programme qui se trouve en ligne 1ØØØ.
Le résultat est affiché par la ligne 17Ø, puis testé en ligne 18Ø, et en ligne 31Ø, si l'on a gagné.

POUR JOUER :
Lancer le programme par RUN ; à l'affichage de la question POT DE DEBUT, on répond en entrant un nombre suffisamment élevé, sur lequel vos mises seront décomptées.
Puis on tape ENTREE, et on entre une première mise (différente de Ø), suivie de ENTREE.
Les trois chiffres tirés sont alors affichés un bref instant, ainsi que la mise faite, et le résultat obtenu (gain ou perte).
On peut ensuite miser de nouveau, faire ENTREE et ainsi de suite.
Pour terminer, il suffit de répondre Ø à la question VOTRE MISE.
Le restant de votre pot initial sera alors affiché.

EXTENSIONS POSSIBLES :
On pourra envisager d'améliorer sérieusement les contrôles, utiliser la couleur et des possibilités graphiques plus élaborées pour simuler davantage un véritable Jackpot.

```
 10 'JACKPOT
 20 SEEDTIME(0)
 30 CLS:SPECIAL:SCALE1,2
 40 INPUT"POT DE DEBUT :";P
 50 CLS:PEN2:PAPER3:SCALE1,2
 60 FOR I=1 TO 3
 70 CURSOR 5,I:PRINT"                          "
 80 NEXT I
 90 PAPER0:CURSOR9,2:PRINT"   ":CURSOR 14,2:PRI
NT"   ":CURSOR 19,2:PRINT"   "
 100 SCALE1,1:CURSOR 0,11:BEEP:INPUT"VOTRE MISE
:";M
 110 IF M=0 THEN GOTO 2000
 120 GOSUB 1000
 130 A=X
 140 GOSUB1000
 150 B=X
 160 GOSUB1000
 170 SCALE1,2:CURSOR10,2:PRINTA:CURSOR 15,2:PRI
NTB:CURSOR 20,2:PRINTX
 180 IF A=B AND A=X THEN GOTO 300
 190 N=-M
 200 SCALE1,1:CURSOR 0,11:PRINT"MISE";M;" === R
ESULTAT ";N
 210 PRINT:INPUT"TAPER 'ENTREE' ... ";R$
 220 P=P+N
 230 GOTO 50
 300 N=M*50
 310 IF A=7 THEN N=M*500
 320 GOTO 200
 999 END
 1000 X=INT(RND(1,10))
 1010 RETURN
 2000 CLS:CURSOR 0,4
 2010 PRINT"VOUS AVEZ $";P
 2020 PRINT:PRINT"A BIENTOT ..."
 2030 SCALE1,1:STANDARD
```

Roulette

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : hasard (plusieurs joueurs).
Difficulté du jeu : facile.

LE JEU :
Une boule se déplace sur un terrain de jeu comportant des cases numérotées de 1 à 36 ainsi que trois cases supplémentaires le ØØ, une case P (comme pair) et une case I (comme impair).
La boule parcourt au moins une fois chaque case, puis s'arrête au deuxième tour, on peut ensuite la relancer pour une autre partie.

LE PROGRAMME :
La boule K (5Ø-11Ø) dessine le terrain de jeu, auquel la ligne 12Ø rajoute les trois symboles P, ØØ et I.
La boucle K (13Ø-2ØØ) réalise le déplacement de la boule, l'arrêt étant contrôlé par les lignes 16Ø et 18Ø.
La ligne 17Ø réalise une légère temporisation.
La ligne 21Ø tire la position d'arrêt.
La ligne 23Ø gère les essais suivants.

POUR JOUER :
Il suffit de faire RUN, et de regarder.
Une fois la boule arrêtée, il suffira de frapper une touche quelconque (sauf "N") pour la relancer. Taper N pour sortir du programme.

EXTENSIONS POSSIBLES :
On pourra essayer de se rapprocher davantage d'une véritable roulette, de différentes manières : mieux dessiner le terrain de jeu, introduire des couleurs, introduire différentes possibilités de pari, et assurer la gestion des gains correspondants, tenir les comptes de plusieurs joueurs, et annoncer les résultats sous une forme plus usuelle.

```
 10 'ROULETTE
 20 SEEDTIME(0)
 30 CLS:SPECIAL:SCALE1,2
 40 CLS
 50 FOR K=1 TO 4
 60 FOR I=1 TO 9
 70 S=I+I
 80 CURSOR 3*I-2,K*3-3:PRINT (S AND K=1) + ((S+
18) AND K=2) + ((S-1) AND K=3) + ((S+17) AND K=
4)
 90 NEXT I
 110 NEXT K
 120 CURSOR 0,2:PRINT"P":CURSOR 0,5:PRINT"OO":C
URSOR 0,8:PRINT"I"
 130 FOR K=1 TO 4
 140 FOR I=1 TO 10
 150 IF I=10 AND K<4 THEN CURSOR 0,K*3 :ELSE CU
RSOR 3*I-1,K*3-2
 160 X=X-1:BEEP
 170 PAUSE .1
 180 IF X=1 THEN PRINT"*":GOTO 230
 185 IF K*I<>40 THEN PRINT"."
 190 NEXT I
 200 NEXT K
 210 X=INT(RND(0,38))
 220 GOTO 130
 230 CURSOR 0,12:INPUT"AUTRE ESSAI ? (O/N) :";R
$
 240 IF R$<>"N" THEN RUN
 250 STANDARD
 260 END
```

Couloirs

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :
Il fait appel à des notions de logique élémentaire. Le joueur peut se présenter à l'entrée, soit du couloir numéro 1, soit du couloir numéro 2. Dans ces deux entrées sont disposés des tourniquets qui basculent à votre passage et vous laissent passer, une fois vers la droite, et la fois suivante vers la gauche. De plus, un tourniquet analogue a été disposé dans le couloir central, pourrez-vous prévoir où (sortie A ou B) va s'effectuer votre prochaine sortie ?

LE PROGRAMME :
Les lignes 3Ø à 7Ø dessinent les couloirs (et les trois tourniquets).
La ligne 1ØØ attend votre choix.
Le décodage du chemin est fait par HECTOR en utilisant trois variables logiques I, J et K, représentant les états des trois tourniquets (lignes 13Ø à 27Ø).

POUR JOUER :
Après la frappe de RUN, le couloir est dessiné sur l'écran, les tourniquets étant visibles, et l'on devra taper 1 ou 2 en réponse à la question de HECTOR (Ø pour finir).
Dans la version retenue ici, la réponse est alors immédiatement affichée, et le chemin suivi depuis le début du jeu, ainsi que les sorties correspondantes, sont affichés sur l'écran. Ceci présente l'avantage de permettre à plusieurs joueurs d'exercer leur sagacité en essayant de prévoir, pour une entrée donnée, ce que va répondre HECTOR.

EXTENSIONS POSSIBLES :
Elles seront très nombreuses ; il est très facile d'imaginer un couloir plus complexe avec d'autres entrées, d'autres sorties et d'autres tourniquets. On peut imaginer un affichage graphique des résultats en dessinant, un bref instant, l'état des tourniquets après la réponse du joueur. On peut aussi ajouter un test des réponses (jeu à plusieurs), décompter les points de chaque joueur, etc.

```
 10 'COULOIRS
 20 CLS:SPECIAL:SCALE1,1:CLS
 30 PRINT" ====================="
 40 PRINT:PRINT" B       ---------   X    2"
 50 PRINT:PRINT" --------   X     ------"
 60 PRINT:PRINT" A       ---------   X    1"
 70 PRINT:PRINT" ====================="
 80 I=0:J=0::K=0
 90 R$="":T$=""
 100 CURSOR 0,11:INPUT"ENTREE PAR 1 OU 2 : ";C$

 105  IF C$="0" THEN GOTO 280
 110 IF LEN(C$)<>1 OR C$<"1" OR C$>"2" THEN BEE
P:GOTO 100
 115 BELL
 120 R$=R$+C$
 130 PRINT:PRINTR$
 140 IF C$="1" THEN GOTO 190
 150 J=NOT(J)
 160 IF J<>0 THEN GOTO 250
 170 S$="B"
 180 GOTO 220
 190 I=NOT(I)
 200 IF I<>0 THEN GOTO 250
 210 S$="A"
 220 T$=T$+S$
 230 CURSOR 0,16:PRINT"SORTIE EN ";S$:PRINT:PRI
NTT$
 240 GOTO 100
 250 K=NOT(K)
 260 IF K<>0 THEN GOTO 170
 270 GOTO 210
 280 STANDARD
```

Echiquier

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : graphisme.
Difficulté du jeu : variable, selon utilisation.

LE JEU :
Le programme proposé vient en appoint du programme Echecs ; il donne, en effet, la possibilité de réaliser un affichage graphique de bonne qualité pour des jeux utilisant un damier de 8 cases (on pourra facilement l'étendre à 1Ø cases sur 1Ø cases). Un repérage a été prévu, horizontalement, par des lettres (de A à H), et, verticalement, par des nombres (de 1 à 8).

LE PROGRAMME :
La ligne 3Ø définit une variable A$ qui contient les éléments de dessin. La ligne 4Ø imprime les lettres horizontales et la première ligne du dessin.
La boucle I (5Ø-14Ø) permet l'impression, le motif à imprimer étant affiché ou non selon la valeur de (I + K)/2 ; la ligne 1ØØ réalise l'impression.

POUR JOUER :
L'ordre RUN provoque l'affichage du damier sur l'écran.
Pour utiliser le programme, il conviendra de tenir compte du repérage des cases utilisables qui sont les centres des carrés noirs ou blancs.

EXTENSIONS POSSIBLES :
On pourra utiliser ce programme pour toute sorte de jeux se déroulant sur un damier (échecs, dames anglaises, le loup et les agneaux, etc.).

```
10 'ECHIQUIER
20 CLS1:PEN0:COVER1
30 A$=CHR$(127)+CHR$(127)+CHR$(127)
40 PRINT"      A   B   C   D   E   F   G   H":PRINT
50 FOR I=1 TO 8
60 FOR J=1 TO 2
70 IF J=2 THEN PRINTI;" ";:ELSE PRINT"    ";
80 PAPER2
90 FOR K=1 TO 8
100 IF (I+K)/2<>INT((I+K)/2) THEN PRINTA$;:ELS
E PRINT"   ";
110 NEXT K
120 PAPER1:PRINT
130 NEXT J
140 NEXT I
150 PEN2:PAPER0:COVER0
```

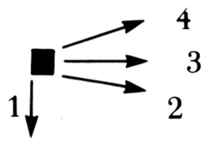
Le jeu de la fève

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexion (deux joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Présentant une certaine analogie avec le jeu de WYTTHOF, cette version offre en fait 4 possibilités de déplacement ; la position de départ est ici fixée en (Ø, Ø), et pourrait éventuellement être modifiée.
Les directions de déplacement disponibles sont : droite, bas, diagonale bas et droite, diagonale haut et droite.
On joue en principe avec la règle : le premier arrivé en (9, 9) a perdu.

LE PROGRAMME :
La ligne 3Ø et la boucle I (4Ø-6Ø) dessinent le terrain de jeu.
La ligne 1ØØ réalise le test de fin de partie.
La ligne 13Ø permet d'exécuter l'un des sous-programmes de déplacement.
Les lignes 14Ø et 15Ø assurent le changement de joueur.

POUR JOUER :
On lance le programme par RUN, et le terrain de jeu s'affiche, ainsi que la question DIRECTION (1, 2, 3, 4) ; les 4 directions proposées correspondent aux déplacements suivants :



Le joueur 1 entre son déplacement, et tape ENTREE.
Le mouvement correspondant est réalisé, puis on obtient l'affichage JOUEUR 2 et DIRECTION.
Le joueur 2 procède de même, et ainsi de suite, jusqu'à ce que l'un des joueurs arrive en (9, 9) auquel cas le jeu se termine.

EXTENSIONS POSSIBLES :
On peut imaginer d'introduire en début de partie des cases pièges (chacun essayant alors de faire venir l'autre sur l'une de ces cases).
On peut aussi jouer avec plusieurs symboles, si possible de couleurs différentes, déplacés simultanément, envisager des terrains de jeu triangulaires, aux dimensions arbitraires, etc.

```
 10 'LE JEU DE LA FEVE
 20 CLS:SPECIAL:SCALE1,1
 30 PRINT"    0 1 2 3 4 5 6 7 8 9"
 40 FOR I=0 TO 9
 50 PRINT:PRINTI;" . . . . . . . . . ."
 60 NEXT I
 70 X=4:Y=0
 80 J=1
 90 CURSOR X,2*Y+2:PRINT"#"
 100 IF X=22 AND Y=9 THEN GOTO 200
 110 CURSOR 0,25:PRINT"JOUEUR";J;:INPUT"DIR. (1
,2,3,4) :";I$
 120 I=VAL(I$):BEEP
 130 ON I GOSUB 1000,2000,3000,4000
 140 J=J+1
 150 IF J=3 THEN GOTO 80
 160 GOTO 90
 200 PRINT:PRINT"LE JOUEUR";J;"A GAGNE ...":LAS
ER
 210 STANDARD
 220 END
 1000 Y=Y+1
 1010 RETURN
 2000 GOSUB 1000
 3000 X=X+2
 3010 RETURN
 4000 GOSUB 3000
 4010 Y=Y-1
 4020 RETURN
```

Jeu de l'oie aléatoire

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : hasard (deux joueurs).
Difficulté du jeu : facile.

LE JEU :
Le jeu reprend la très ancienne idée du jeu de l'oie, mais en y ajoutant la possibilité offerte par l'ordinateur de choisir au hasard les cases "à problème".
La règle du jeu est sans grand intérêt, puisque les joueurs n'interviennent à aucun moment dans le déroulement de la partie, qui est entièrement gérée par HECTOR ; il suffira d'appuyer sur une touche après chaque lancer de dé, et de regarder ce qui se passe.

LE PROGRAMME :
La ligne 6Ø permet d'appeler le sous-programme 1ØØØ à 11ØØ qui réalise l'affichage du terrain de jeu, le contrôle de la position des joueurs et l'affichage de cette position. Cet affichage sera d'ailleurs refait en cours de partie, pour tenir compte du déplacement réellement effectué (qui n'est pas nécessairement celui indiqué par le dé).
La ligne 7Ø s'occupe du changement de joueur, et la ligne 8Ø effectue le lancer de dé.
La ligne 9Ø traite le cas où l'un des joueurs termine.
La ligne 13Ø tire un nombre aléatoire entre Ø et 3, qui servira s'il y a lieu pour déterminer le vrai déplacement du joueur, en envoyant dans l'un des sous-programmes de "chance".
Les lignes 1ØØØ et 1Ø1Ø font avancer les pions, et les lignes 1Ø2Ø à 1Ø8Ø redessinent à chaque affichage le terrain de jeu en entier.

POUR JOUER :
On lancera le jeu par RUN. Le terrain de jeu est alors dessiné, puis on voit apparaître l'affichage JOUEUR 1 :...
On devra alors appuyer sur une touche quelconque pour que le coup du joueur numéro 1 soit pris en compte ; ne pas s'inquiéter du temps de réaction, qui pourra sembler un peu long.
Le déplacement indiqué par les dés sera exécuté, puis au bout de quelque temps, un second déplacement sera éventuellement effectué : c'est le déplacement "CHANCE". On obtiendra alors l'affichage JOUEUR 2 :...
Le joueur 2 appuie à son tour sur une touche quelconque et ainsi de suite...
Le joueur gagnant sera signalé par un affichage BRAVO en fin de partie.
On peut aussi s'arrêter à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
On peut agrandir le terrain de jeu, introduire davantage d'aléas, changer leur nature. On pourrait aussi choisir les cases "chance" en début de jeu, ajouter des affichages en conséquence, etc.

```
10 'JEU DE L'OIE ALEATOIRE
20 CLS:SPECIAL:SCALE1,2
30 SEEDTIME(1)
50 X=0:Y=0:J=1:I=X
60 GOSUB1000
70 J=ABS(J-1)
80 I=INT(RND(1,7))
90 IF X=25 OR Y=25 THEN LASER:CURSOR 0,12:PRIN
T"===  BRAVO ...":STANDARD:END
100 BEEP:CURSOR0,7:PRINT"JOUEUR";J+1;"COUP";I
110 INPUT"TAPER ENTREE :";R$
120 GOSUB 1000
130 I=INT(RND(0,4))
140 F=INT(RND(1,6))
150 ON F GOSUB 200,300,400,500,160
160 GOTO 60
200 I=I+3
210 RETURN
300 I=-I
310 RETURN
400 I=2*I
410 RETURN
500 IF J=0 THEN X=INT(RND(0,20)):RETURN
510 Y=INT(RND(0,20)):RETURN
1000 IF J=0 THEN X=X-I*(X+I<=25)
1010 Y=Y-J*I*(Y+I<=25)
1020 CURSOR0,0:PRINT"ABCDEFGHIJKLMNOPQRSTUVWXY
Z"
1030 FOR I=0 TO 25
1040 CURSOR I,2:IF I<=X THEN PRINT" 1":ELSE IF
I<>X+1 THEN PRINT"  "
1050 CURSOR I,4:IF I<=Y THEN PRINT" 2":ELSE IF
I<> Y+1 THEN PRINT"  "
1060 BEEP
1070 NEXT I
1080 RETURN
```

Niveau 2

Nom du programme	*Thèmes et particularités*
Dé truqué	A utiliser, si vous êtes malhonnête, dans d'autres programmes.
Visions	Un jeu sur les mots, pour l'ordinateur.
421	Une simulation simplifiée, plusieurs joueurs.
L'électricien fou	Logique et difficile, pour un seul joueur.
La course au 20	Jeu de réflexion, pour deux joueurs ou plus.
Musique	Révisez vos notions de solfège, grâce à l'ordinateur.
Poker	Une base de départ, pour mettre au point votre programme.
Le jeu de saute-mouton	Un jeu de réflexion, pour deux joueurs.
Lettres	Un jeu sur les mots, assez simple.
Machinamot	Laissez l'ordinateur créer des vocables nouveaux.
Espion	Dites-lui votre amour en code.
Accrochez les wagons	Pour les plus jeunes ; très visuel.
Tirage d'un dé truqué	Révisez vos connaissances en statistiques.
Tour de cartes	Votre ordinateur a plus d'un tour dans son sac.
Le jeu de Marienbad	Un jeu de logique pour deux joueurs, dans une version étendue.
L'indésirable	Vous êtes propriétaire. Arriverez-vous à mettre la main sur ce locataire indésirable ?
Jour de la semaine	Profitez-en pour faire le point sur les anniversaires.
Le jeu de la vie	Un grand classique, en deux versions : manuelle et automatique.

Dimensionnement d'un tableau Recherche d'un élément dans un tableau et dans une chaîne L'instruction DATA

L'INSTRUCTION DIM OU INSTRUCTION DE DIMENSIONNEMENT D'UN TABLEAU

Les structures de tableaux sont très couramment utilisées en programmation, se révélant même indispensables dans un grand nombre de cas. Si vous voulez, par exemple, entrer vos recettes et dépenses pour tous les jours du mois (on supposera pour simplifier des mois de trente jours) vous pourriez définir 30 variables appelées PREMIER JOUR, DEUXIEME JOUR, etc. Il est plus facile, et plus rationnel d'utiliser une boucle I, I variant de 1 à 3Ø, et une variable R (I) pour les recettes, et D (I) pour les dépenses ; ce faisant, vous avez, sans le savoir, utilisé une structure de tableau à une dimension ; ceci ne pourra être admis par HECTOR que si vous l'avez auparavant prévenu que vous alliez utiliser un tel tableau, en donnant une instruction DIM R (3Ø) pour les recettes, et DIM D (3Ø) pour les dépenses.

On pourrait aussi imaginer de conserver dépenses et recettes dans un même tableau, en lui donnant cette fois deux dimensions : par exemple DIM RD (3Ø, 2) (ou DIM RD (2, 3Ø)). Ainsi, avec DIM RD (3Ø, 2) on pourra convenir que RD (I, 1) représente les recettes, et RD (I, 2) les dépenses. Des tableaux de plus de 2 dimensions sont également possibles.

On peut aussi concevoir des tableaux dont les éléments sont des chaînes de caractères, et non plus des nombres, avec 255 caractères au maximum.

Ainsi DIM NOM$ (12) réserve 12 emplacements en mémoire, pour des mots de 255 caractères au maximum.

Sur HECTOR il n'est pas obligatoire de dimensionner un tableau qui contient moins de 10 éléments.

LA RECHERCHE D'UN ELEMENT DANS UN TABLEAU : R (I)

Si l'on a un tableau à une dimension, numérique ou alphanumérique, défini par une instruction comme DIM R (30) ou DIM N$ (20), le Ième élément de ce tableau est repéré par R (I) (ou N$ (I)) ; il sera alors manipulé exactement comme une variable ordinaire (affectation, calcul, tests, etc.).

Ainsi INPUT R (3) permet d'entrer à la main la valeur du 3ᵉ élément numérique du tableau R. PRINT R$ (5) permet l'affichage du 5ᵉ élément du tableau alphanumérique R$, etc.

LA RECHERCHE D'UN ELEMENT DANS UNE CHAINE : MID$

Pour extraire le Ième élément de la chaîne alphanumérique A$, on utilisera l'instruction MID$.

Ainsi, si A$ = "BONJOUR", MID$ (A$, 3, 1) donne le caractère N ; de plus MID$ (A$, 3, 3) donne la sous-chaîne "NJO".

LE TRAITEMENT DE DONNEES MULTIPLES INSTRUCTIONS READ, DATA, RESTORE

Pour remplir facilement un tableau, numérique ou alphanumérique, il sera intéressant d'utiliser des lignes DATA, qui contiennent les données inscrites les unes à la suite des autres, séparées par une virgule.

Ainsi, on pourra écrire :

DATA LEON, JULES, 13, rue MARTIN...

Ces données peuvent être lues dans des boucles FOR... NEXT, au moyen de l'instruction READ. On peut recommencer la lecture au début des DATA, à l'aide de l'instruction RESTORE.

Dé truqué

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

LE JEU :
On veut essayer de fabriquer un dé truqué selon sa propre convenance. La méthode proposée ici consiste à définir au départ une probabilité de réalisation pour chaque type de tirage (soit la probabilité d'obtenir un 1, un 2, un 3, etc. que chacune de ces probabilités soit comprise entre Ø et 1, et que la somme de toutes les probabilités soit un nombre très voisin de 1).

LE PROGRAMME :
Il utilise le générateur aléatoire de HECTOR supposé de qualité convenable (on pourra aisément utiliser tout autre générateur, si besoin est). La boucle I (7Ø-14Ø) s'occupe de l'entrée de chacune des probabilités, avec vérification à chaque entrée et sur le total (ligne 11Ø) ; chaque probabilité est conservée dans une variable indicée E (I) : ligne 9Ø du programme.
Le lancer de dé truqué est effectué à l'aide d'une autre boucle I (18Ø-2ØØ), la ligne permettant le truquage se trouvant en 19Ø. Un nouveau tirage pourra être effectué sur pression de ENTREE (ligne 22Ø).

POUR JOUER :
Après le lancement de l'exécution par RUN, on obtient l'affichage de la question PROBABILITE DU DE 1, à laquelle on devra répondre par un nombre compris entre Ø et 1. Il conviendra de ne pas oublier que le total de toutes les probabilités entrées ne devra en aucun cas dépasser 1.1 (tolérance de Ø.1).
Des messages d'erreurs pourront être affichés et le programme sera éventuellement relancé. Lorsque toutes les probabilités auront été définies, le premier tirage sera automatiquement effectué, et il suffira de presser la touche ENTREE pour effectuer les tirages suivants ou de taper F pour terminer. On arrête le programme en tapant CNT et C.

EXTENSIONS POSSIBLES :
Il sera évidemment bien tentant d'utiliser sournoisement ce petit programme pour améliorer vos chances de gain dans un jeu de hasard utilisant des lancers de dés. On pourra alors envisager de figer les données définissant le dé truqué dans une variable A$, etc. On pourra aussi envisager de présenter les résultats correspondant à une série de lancers de dés sous une forme spectaculaire, qui mette le truquage en évidence...

```
 10 'DE TRUQUE
 20 CLS
 30 SPECIAL:SCALE 1,2
 40 DIM E(6)
 50 P=0
 60 CLS
 70 FOR I=1 TO 6
 80 CURSOR 17,0:PRINT"          "
 90 CURSOR 0,0:PRINT"PROBABILITE DU DE";I;:INPU
T":";E(I)
 100 P=P+E(I)
 110 IF E(I)<0 OR E(I)>1 OR P>1.1 THEN GOTO 240

 120 CURSOR4*I-2,2:PRINTI
 130 CURSOR4*I-3,4:PRINTE(I)
 140 NEXT I
 150 IF P<.9 THEN BEEP:GOTO50
 160 PAUSE 1
 170 N=INT(RND(1,7))
 180 FOR I=1 TO 6
 190 IF N=I AND RND(0,1)>E(I) THEN I=6:NEXT I:G
OTO 170
 200 NEXT I
 210 CURSOR 0,7:PRINT"TIRAGE .. :";N
 220 BEEP:PRINT:PRINT"TAPER 'ENTREE' POUR RETIR
ER":INPUT"       'F' POUR TERMINER :";R$
 230 IF R$="F" THEN STANDARD:END:ELSE GOTO 170
 240 CURSOR 0,12:PRINT"IMPOSSIBLE : 0 <= P <= 1
"
 250 SIREN
 260 P=P-E(I)
 270 PAUSE1.5
 280 CURSOR 0,12:PRINT"
"
 290 HUSH
 300 GOTO 90
```

Visions

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : dessin.
Difficulté du jeu : facile.

LE JEU :
HECTOR vous permet, pour la première fois, d'entrer dans son intimité en vous laissant entrevoir sa propre représentation du monde ! Vous allez, en effet, lui proposer des mots, n'importe lesquels, et il va vous répondre en vous proposant une représentation graphique de ce que représente ce mot pour lui. Le jeu sera beaucoup plus amusant si vous le pratiquez à plusieurs, et vous pourrez quelquefois avoir des surprises.

LE PROGRAMME :
Une boucle I (4Ø-12Ø) se charge de réaliser les dessins.
La ligne 5Ø permet de transformer en nombre chaque lettre du mot entré au début (ligne 3Ø). Ce nombre sera ensuite traduit en base 2 par une boucle J (lignes 6Ø à 11Ø) ; selon le résultat de cette transformation, un carré sera imprimé, ou non (ligne 9Ø).

POUR JOUER :
On débute l'exécution du programme par RUN, ce qui amène l'affichage de la question QUEL DESSIN. On répond en entrant un mot quelconque qui sera transformé en dessin par HECTOR dès que l'on aura frappé ENTREE.
Pour recommencer, il suffira de frapper une touche quelconque, et CNT et C pour finir. Je vous laisse savourer le dessin qu'il a réalisé lorsqu'il a appris le prénom de l'auteur de ce livre.

EXTENSIONS POSSIBLES :
La même idée peut être reprise en introduisant des éléments de symétrie dans les dessins, en ajoutant la couleur. On peut aussi imaginer de coder par cette méthode un certain nombre de dessins, que l'on introduirait au fur et à mesure dans la machine et l'on pourra alors créer un jeu dans lequel HECTOR deviendra de plus en plus savant. On peut aussi réaliser un jeu à deux joueurs dans lequel HECTOR vous demande de trouver le mot qui correspond à un certain dessin, etc.

```
10 'VISIONS
20 CLS:SPECIAL:SCALE 1,1
30 INPUT"QUEL DESSIN :";R$
35 IF LEN(R$)=0 THEN STANDARD:END
40 PAPER 1:FOR I=1 TO LEN(R$)
50 N=9*ASC(MID$(R$,I,1))-340
60 J=1
70 S=N/2-INT(N/2)
80 N=INT(N/2)
90 IF S<>0 THEN CURSOR J+5,I+3:PRINTCHR$(127)
100 J=J+1
110 IF J<10 THEN GOTO 70
120 NEXT I:PAPER 0
130 CURSOR 0,25:INPUT"TAPER 'ENTREE' :";R$
140 GOTO 20
```

421

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : moyenne.

LE JEU :
Chacun des joueurs devra, à son tour, lancer 3 dés, et peut ensuite choisir, sur les deux coups suivants, s'il désire relancer ou non 1, 2 ou 3 dés. Il marque alors un score correspondant au nombre indiqué par les dés, suivant les règles choisies, et le joueur suivant procède de la même manière.

LE PROGRAMME :
Le nombre de joueurs est tout d'abord entré en ligne 4Ø.
La ligne 13Ø affiche le numéro de joueur et le nombre d'essais déjà effectués.
Un sous-programme placé en ligne 4ØØØ, utilisé 3 fois dans la boucle J (2ØØ-25Ø), réalise le lancer de dés.
Les retirages sont effectués par les lignes 1ØØØ, 2ØØØ et 3ØØØ.
La variable R$ gère les 2e et 3e tirages. L'impression du résultat est réalisée dans tous les cas par les lignes 14Ø-15Ø.
Les tests pour le nombre d'essais et pour le changement de joueur sont effectués aux lignes 26Ø et 28Ø.

POUR JOUER :
On lance l'exécution du programme par RUN ce qui conduit à l'affichage NOMBRE DE JOUEURS. On entre le nombre de joueurs, et l'on tape ENTREE. L'écran s'efface et affiche JOUEUR N° 1. ESSAI A, PARTIE P, ainsi que le tirage réalisé, et les noms des dés (A, B, C).
Si l'on ne désire pas refaire de tirage, il suffit de frapper ENTREE. La main passe alors au joueur suivant.
Pour effectuer un autre tirage, il suffit de taper le nom du ou des dés à retirer, et de faire ENTREE : par exemple, pour retirer A et C on tapera AC suivi de ENTREE.
Les joueurs suivants procèderont de la même façon.
Taper FIN pour arrêter le jeu.

EXTENSIONS POSSIBLES :
La plus évidente consiste à assurer la gestion des résultats des joueurs, en se référant aux règles de gain usuelles du jeu du 421. On pourra envisager l'affichage après chaque tour d'un tableau récapitulant les résultats des joueurs.
On pourra aussi améliorer l'affichage des dés (se référer aux divers programmes d'affichage des dés se trouvant dans le même ouvrage).

```
 10 '4 2 1
 20 SEEDTIME(0)
 30 CLS:SPECIAL:SCALE 1,2
 40 INPUT"NOMBRE DE JOUEURS :";N
 50 P=1
 60 I=1
 70 K=0
 80 CLS
 90 R$="ABC"
 100 GOTO 200
 110 K=K+1
 120 CURSOR 4,0:PRINT"===   4 2 1   ==="
 130 CURSOR 0,3:PRINT"JOUEUR";I;"ESSAI";K;"PART
IE";P
 140 CURSOR5,6:PRINTA:CURSOR12,6:PRINTB:CURSOR1
9,6:PRINTC
 150 CURSOR6,8:PRINT"A":CURSOR13,8:PRINT"B":CUR
SOR20,8:PRINT"C"
 160 R$="":CURSOR 0,11:PRINT"
      "
 170 CURSOR 0,11:INPUT"==  ABC ? :";R$
 175 IF R$="FIN" THEN GOTO 310
 180 IF R$="" THEN GOTO 270
 190 R$=R$+"DDD"
 200 FOR J=1 TO 3
 210 GOSUB 4000
 220 X$=MID$(R$,J,1)
 230 X=ASC(X$)-64
 240 ON X GOSUB 1000,2000,3000,4000
 250 NEXT J
 260 IF K<3 THEN GOTO 110
 270 I=I+1
 280 IF I<=N THEN GOTO 70
 290 P=P+1
 300 GOTO 60
 310 STANDARD
 320 END
 1000 A=D
 1010 RETURN
 2000 B=D
 2010 RETURN
 3000 C=D
 3010 RETURN
 4000 D=INT(RND(1,7))
 4010 RETURN
```

L'électricien fou

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez difficile.

LE JEU :
Un immeuble comportant 1Ø fenêtres, numérotées de Ø à 9, apparaît sur l'écran. Le joueur devra à chaque essai manœuvrer l'un des interrupteurs, en indiquant son numéro (de Ø à 9).
En principe, la fenêtre correspondante devrait s'éteindre, mais il est aussi possible qu'une autre s'allume (ou s'éteigne). Il s'agit, bien sûr, d'essayer d'obtenir l'extinction des 1Ø fenêtres, avec le moins possible d'essais.

LE PROGRAMME :
Une table A, en ligne 5Ø, contiendra l'état (allumé ou éteint) des 1Ø fenêtres.
La boucle I (7Ø-9Ø) et l'autre boucle I (11Ø-16Ø) dessinent l'immeuble dans son état initial.
La ligne 17Ø regarde si tout est éteint, auquel cas l'on ira en ligne 31Ø afficher le nombre d'essais.
La ligne 18Ø attend l'indication de l'interrupteur à manœuvrer.
Les lignes 19Ø à 29Ø définissent le résultat de la manœuvre.

POUR JOUER :
RUN provoque l'affichage de l'immeuble, avec toutes les fenêtres allumées.
A la question INTERRUPTEUR NUMERO, on répond en indiquant le numéro de l'interrupteur à manœuvrer, suivi de ENTREE.
Le résultat de la manœuvre est alors affiché sur l'écran (des fenêtres s'allument ou s'éteignent).
On continue ainsi jusqu'à ce que toutes les fenêtres soient éteintes, et l'on obtient alors l'affichage du nombre d'essais réalisés.

EXTENSIONS POSSIBLES :
On pourra envisager un graphisme très amélioré avec si possible l'utilisation de la couleur. On pourra aussi définir un jeu à deux joueurs, augmenter le nombre de fenêtres, modifier la formule définissant l'allumage ou l'extinction...

```
10 'L'ELECTRICIEN FOU
20 SEEDTIME(0)
30 CLS:SPECIAL:SCALE 1,2
40 M=0
50 DIM A(10)
60 CLS
70 FOR I=1 TO 6
80 PAPER1:CURSOR 0,I:PRINT"
    "
90 NEXT I
100 PAPER0:T=0
110 FOR I=1 TO 10
120 T=T+A(I)
130 PAPER 0:IF A(I)=0 THEN PAPER 3
140 CURSOR 2*I+1,3:PRINT" "
150 PAPER 0:CURSOR 2*I,5:PRINTI-1
160 NEXT I
170 IF T=10 THEN GOTO 310
180 CURSOR 0,11:BEEP:INPUT"INTERRUPTEUR :";N
190 N=N+1
200 M=M+1
210 IF A(N)-1 THEN GOTO 290
220 A(N)=1
230 R=TAN(RND(0,1)+N/RND(0,1)-N)-SIN(RND(0,1)/
N)+336*SIN(8*N)
240 N=INT(10*(R-INT(R)))
250 IF N=0 THEN N=INT(RND(1,11))
260 IF A(N)=1 THEN GOTO 290
270 A(N)=1
280 GOTO 100
290 A(N)=0
300 GOTO 230
310 PRINT:PRINT"TROUVE EN";M;"ESSAIS."
320 STANDARD
```

La course au 20

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : logique (deux joueurs, ou plus).
Difficulté du jeu : moyenne.

LE JEU :
Il est possible d'imaginer pour ce jeu diverses règles, et nous n'avons retenu ici que la règle usuelle pour deux joueurs, qui en fait un jeu apparenté aux jeux de déplacement, ou d'allumettes : 4 tas de 2Ø étoiles vont apparaître en début de partie, et chaque joueur peut prendre autant d'étoiles qu'il le désire ; mais ceci dans un seul tas à la fois. Celui qui prend la dernière étoile est considéré comme perdant.

LE PROGRAMME :
La boucle I (3Ø-5Ø) dessine 4 rangées de 2Ø étoiles.
La variable F, dimensionnée en ligne 7Ø, servira à enregistrer le nombre d'étoiles prises dans chaque rangée.
La variable T contiendra le nombre total d'étoiles enlevées (ligne 6Ø).
La ligne 12Ø va contrôler qu'il reste suffisamment d'étoiles sur la ligne indiquée.
Les boucles I (15Ø-19Ø) et J (16Ø-18Ø) suppriment les étoiles ; la ligne 22Ø teste la fin de partie.

POUR JOUER :
Lancer le programme par RUN ; le terrain de jeu s'affiche, puis les inscriptions JOUEUR 1 et LIGNES, puis CASES. On répond en entrant un numéro de ligne (entre 1 et 4) suivi de ENTREE, et le nombre d'étoiles à enlever suivi de ENTREE. Les étoiles sont alors supprimées, et c'est au tour du deuxième joueur.
La partie se termine, lorsque la dernière étoile a été enlevée, par l'affichage du joueur gagnant.

EXTENSIONS POSSIBLES :
Outre une amélioration de la qualité du graphisme et des contrôles, de nombreuses variantes sont envisageables en modifiant, si besoin est, les règles de départ : on peut autoriser l'enlèvement d'un nombre impair d'étoiles seulement ou alterner pair et impair ou tirer au hasard le nombre à enlever ou encore jouer à plusieurs...

```
10 'LA COURSE AU 20
20 CLS:SPECIAL:SCALE 1,2
30 FOR I=1 TO 4
40 CURSOR 0,2*I-2:PRINTI;" *******************
*"
50 NEXT I
60 T=0
70 DIM F(4)
80 B=1
90 CURSOR 0,9:PRINT"JOUEUR ... :";B
100 INPUT"LIGNE .... :";L
110 INPUT"CASES .... :";C
120 IF C+F(L)>20 THEN BEEP:GOTO 90
130 F(L)=F(L)+C
140 T=T+C
160 FOR J=F(L)-C TO F(L)
170 CURSOR 24-J,2*L-2:PRINT" ":BEEP
180 NEXT J
200 B=B+1
210 IF B=3 THEN B=1
220 IF T<80 THEN GOTO 90
230 PRINT"LE JOUEUR";B;"A GAGNE ..."
235 LASER
240 STANDARD
```

Musique

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu éducatif.
Difficulté du jeu : moyenne.

LE JEU :
Il se présente comme un exercice d'apprentissage du solfège : l'appareil annonce une note (en clé de sol), suivie de la mention GRAVE ou AIGUE, selon sa place sur la portée. Il faudra répondre en indiquant sur quelle ligne la note sera affichée.
La note est alors dessinée sur la portée, suivie d'un message BRAVO ou PERDU, selon que l'on avait trouvé juste ou non.

LE PROGRAMME :
La boucle I (5Ø-7Ø) dessine la portée et affiche les numéros des lignes.
La ligne 13Ø définit le nom des notes, et la ligne 14Ø réalise le tirage aléatoire d'une note.
Le nom de la note est calculé à la ligne 15Ø et il est affiché par la ligne 16Ø, suivi de la mention GRAVE ou AIGUE, selon le cas.
La réponse sera entrée en ligne 19Ø.
La ligne 2ØØ s'occupe du décodage de la réponse.
La ligne 21Ø affiche la note.
La ligne 24Ø regarde si l'on a gagné ou perdu et affiche le message correspondant.
Les lignes 25Ø et 26Ø permettent de faire un nouvel essai.

POUR JOUER :
On lance le programme par RUN.
On obtient alors l'affichage de la portée, ainsi que le nom de la note à placer et la question NUMERO DE LA LIGNE.
On répond en donnant un numéro de ligne (de 1 à E) suivi de ENTREE.
La note est alors dessinée et le message BRAVO ou PERDU apparaît, dans ce cas, à l'écran.
On doit taper la touche ENTREE pour continuer, et s'arrêter en tapant F.

EXTENSIONS POSSIBLES :
On peut étendre à une autre clé que la clé de sol, on peut introduire de la couleur, des sons, etc.

```
 10 'MUSIQUE
 20 SEEDTIME(0)
 30 STANDARD:PEN2
 40 CLS:PRINT
 50 FORI=1TO14:IF I<10 THENPRINT"    ";I;" ";:EL
SE PRINT"    ";CHR$(55+I);"  ";
 60 IF I=4 OR I=6 OR I=8 OR I=10 OR I=12 THEN F
OR J=1 TO 25 :PRINT"-";:NEXT J:PRINT : ELSE PRI
NT
 70 NEXT I
 130 N$="SI LA SOL FA MI RE DO "
 140 X=INT(RND(1,15))
 150 T=X-1-7*INT((X-1)/7)
 160 C=0:L=16:GOSUB1000:PRINT"      NOTE : ";MID
$(N$,3*T+1,3);
 170 IF X<=7 THEN PRINT" AIGU. ":ELSE PRINT" GR
AVE."
 190 PRINT:INPUT"NUMERO DE LA LIGNE :";L$
 200 Y=-(ASC(L$)-48)*(L$<":")-(ASC(L$)-55)*(L$>
="A")
 210 PEN 1:C=19:L=X-2:GOSUB1000:PRINT"!":L=X-1:
GOSUB1000:PRINT"!":C=18:L=X:GOSUB 1000:PRINT"O"
:PEN2
 220 C=25:L=18:GOSUB1000:
 240 IF Y=X THEN PRINT"BRAVO ...":ELSE PRINT"PE
RDU ..."
 250 PRINT:PRINT"TAPER 'ENTREE' POUR CONTINUER"
:INPUT "       'F' POUR TERMINER        :";R$
 260 IF R$<>"F" THEN GOTO 40
 270 END
 1000 CURSOR C*6+9,226-L*10
 1010 RETURN
```

Poker

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu contre HECTOR.
Difficulté du jeu : facile.

LE JEU :
C'est une simulation d'une partie de poker traditionnelle, effectuée avec un jeu de 32 cartes, dans laquelle le joueur aura, pour partenaire, HECTOR.
Le joueur recevra 5 cartes, repérées par des numéros de 1 à 5, et pourra décider de changer certaines cartes en annonçant celles qu'il écarte. Sa nouvelle donne sera alors affichée, et c'est ensuite au tour de HECTOR de jouer.

LE PROGRAMME :
Les lignes 5Ø et 6Ø définissent les cartes. Une variable N est dimensionnée à 16, en ligne 4Ø, pour recevoir 1Ø cartes joueur et 5 cartes HECTOR.
Le tirage sera effectué par une boucle I (8Ø-14Ø) et le contrôle par une boucle J (11Ø-13Ø).
L'affichage des cartes est réalisé par le sous-programme 5ØØ-61Ø.
Une autre boucle I (19Ø-21Ø) réalise le deuxième tirage du joueur, la ligne 24Ø permettant le deuxième affichage, et la boucle I (5ØØ-57Ø) où l'on entre directement après, réalisera l'affichage du jeu machine.

POUR JOUER :
L'ordre RUN lance l'exécution du programme, et les 5 premières cartes du joueur sont alors affichées.
A la question CARTES REFUSEES, le joueur peut répondre Ø suivi de ENTREE s'il veut arrêter la partie. Pour jouer, il répondra simplement ENTREE s'il s'estime servi, ou il entrera les numéros des cartes qu'il refuse suivis de ENTREE.
Par exemple, s'il tape 235 et ENTREE, son nouveau jeu conservera les cartes 1 et 4. Son jeu sera alors affiché ainsi que celui de HECTOR, ce dernier n'ayant pas, dans ce programme, la possibilité de reprendre des cartes.
Il suffira ensuite de faire ENTREE pour démarrer une autre partie.

EXTENSIONS POSSIBLES :
Elles seront, bien sûr, très nombreuses, depuis un affichage graphique en couleur des cartes jusqu'à l'amélioration du jeu de HECTOR, l'introduction des règles sur les paris et même, pourquoi pas, des possibilités de bluff.

```
 10 'POKER
 20 SEEDTIME(0)
 30 CLS:SPECIAL:SCALE1,2
 40 DIM N(16)
 50 V$="7 8 9 10VADAROAS"
 60 C$="TRCACOPI"
 70 CLS
 80 FOR I=1 TO 16
 90 N(I)=INT(RND(1,33))
 100 IF I=1 THEN GOTO 140
 110 FOR J=1 TO I-1
 120 IF N(I)=N(J) THEN J=I:NEXT J:GOTO 90
 130 NEXT J
 140 NEXT I
 150 K=0:K1=0:P=0
 160 PEN 2:CURSOR0,10:PRINT"CARTES REFUSEES :"
 170 GOSUB 500
 180 IF R$="0" THEN STANDARD:PEN 2:END
 190 FOR I=1 TO LEN(R$)
 200 N(VAL(MID$(R$,I,1)))=N(I+5)
 210 NEXT I
 220 CLS
 230 P=1
 240 GOSUB 500
 250 K=K+5:K1=K1+10
 500 FOR I=1 TO 5
 510 J=2*INT(N(I+K1)/8)+1
 520 F=2*(N(I+K1)-8*INT(N(I+K1)/8))+1
 530 R$=MID$(V$,F,2)
 540 PEN 3+2*(K=5):CURSOR 5*I-4,K+I-1:PRINTI:BE
EP
 550 CURSOR 5*I-5,K+I+1:PRINTR$;":";MID$(C$,J,2
)
 560 CURSOR 5*I-5,K+I+2:PRINT"-----"
 570 NEXT I
 580 IF P=1 THEN P=0:RETURN
 590 IF K=5 THEN PEN 2:CURSOR 0,13:INPUT"TAPER
'ENTREE' :";R$:GOTO 70
 600 CURSOR 18,10:INPUT"";R$
 610 RETURN
```

Le jeu de saute-mouton

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : facile.

LE JEU :
Bien que l'idée en soit très simple, il ne sera pas forcément évident de découvrir une stratégie convenable. On dispose de 3 moutons noirs et de 3 moutons gris, représentés par des symboles graphiques, qui occupent 8 cases ; les moutons peuvent se déplacer d'une case à une case libre. Plusieurs règles pourront intervenir, selon votre propre estimation de l'intérêt de telle ou telle : soit l'on contraint les moutons à se placer sur la première case disponible, en sautant éventuellement par-dessus les moutons adverses, soit on leur laisse choisir leur case d'arrivée, ce qui peut rendre les parties interminables, à vous de choisir.

LE PROGRAMME :
La boucle I (5Ø-8Ø) initialise une table A (I), qui contient les positions des moutons. La boucle I (1ØØ-15Ø) dessine le terrain de jeu et les moutons. Les lignes 17Ø et 18Ø s'occupent du texte et des entrées.
La ligne 19Ø contrôle la validité du coup. Le changement de joueur est réalisé en ligne 21Ø.

POUR JOUER :
On commence l'exécution du programme par RUN. On obtient alors un affichage JOUEUR 1, qui indique le numéro du joueur, suivi de la question DEBUT, FIN.
On répond en indiquant le numéro du pion à déplacer suivi de ENTREE, puis le numéro de la case d'arrivée, suivi de ENTREE.
Le déplacement est alors réalisé ; puis le deuxième joueur procède de la même façon. Le gagnant est en principe celui qui réussit à placer ses trois moutons côte à côte sur les emplacements initialement occupés par l'adversaire.

EXTENSIONS POSSIBLES :
On commencera par rajouter un contrôle du joueur gagnant.
On peut également améliorer la présentation en dessinant de vrais moutons, rendre le jeu plus attractif en les numérotant, et en imposant un certain ordre à l'arrivée.
Enfin, on pourra définir une stratégie pour l'ordinateur, en figeant les règles.

```
10 'LE JEU DE SAUTE-MOUTON
20 CLS:SPECIAL:SCALE1,3
30 DIM A(8)
40 A(1)=0:A(8)=0
50 FOR I=1 TO 3
60 A(I+1)=1
70 A(I+4)=-1
80 NEXT I
90 H=1
100 FOR I=1 TO 8
110 PEN 2:CURSOR3*I-1,3:PRINTI:BEEP
115 CURSOR3*I,2:PRINT":"
120 CURSOR3*I,1:PRINT" "
130 IF A(I)=1 THEN CURSOR 3*I,1:PEN 3:PRINT">"

140 IF A(I)=-1 THEN CURSOR 3*I,1:PEN 1:PRINT"<
"
150 NEXT I
160 J=1-(H=-1)
170 PEN 2:SCALE 1,2:CURSOR0,9:PRINT"JOUEUR";J
180 PRINT:INPUT"DEBUT,FIN :";D,F
185 IF D=0 AND F=0 THEN GOTO 230
190 IF A(D)<>H OR A(F)<>0 THEN BELL:GOTO 170
200 A(D)=0:A(F)=H
210 H=-H
220 SCALE 1,3:GOTO 100
230 STANDARD:PEN 2
```

Lettres

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :
La règle est fortement inspirée d'un jeu de lettres très connu en France, pratiqué notamment dans une émission télévisée (les chiffres et les lettres). HECTOR choisit au hasard un certain nombre de symboles (voyelles et consonnes) dans l'alphabet latin, et chacun des joueurs devra indiquer la longueur du mot qu'il a réussi à obtenir (l'appareil ne vérifiera pas si le mot est correct !).
Le score total de chaque joueur est affiché ensuite.

LE PROGRAMME :
La boucle I (9Ø-12Ø) choisit dans l'alphabet entre 7 et 12 lettres, en fonction de la variable aléatoire L, tirée en ligne 6Ø. Pour éviter d'avoir trop de consonnes, on utilise en ligne 11Ø une petite astuce, qui consiste à refaire au hasard un tirage dans la table A$, contenant des voyelles (on peut en avoir plus ou moins en utilisant d'autres RND).
La boucle 13Ø-18Ø demande à chacun des joueurs la longueur de son mot. Les lignes 19Ø et 2ØØ comparent les résultats et totalisent le score de chaque joueur, affiché par la ligne 21Ø.

POUR JOUER :
On lance l'exécution du jeu avec RUN.
Les lettres sont alors affichées, puis la question JOUEUR 1 LONGUEUR à laquelle le premier joueur répond en donnant la longueur du mot qu'il a obtenu (l'autre joueur devra toutefois vérifier si le mot est correct) ; puis le joueur 2 fera de même et le score de chaque joueur sera alors affiché. Pour continuer la partie, il suffira d'appuyer sur la touche ENTREE ou F pour finir.

EXTENSIONS POSSIBLES :
On pourra bien entendu songer à introduire une temporisation avant de demander leurs réponses aux joueurs.
On pourra aussi entrer non point la longueur du mot, mais le mot lui-même, l'adversaire devant alors répondre à la question CORRECT OU NON ? On peut aussi étendre le jeu à plus de deux joueurs. Et pourquoi ne pas essayer de définir un jeu contre HECTOR, auquel on aurait auparavant entré une liste de mots assez longue, dont il afficherait les lettres dans le désordre, avec éventuellement quelques lettres de complément choisies au hasard ?

```
 10 'LETTRES
 20 CLS:SPECIAL:SCALE1,2
 30 SEEDTIME(0)
 40 S=0:J=0
 50 CLS
 60 L=INT(RND(6,10))
 70 A$="AEIOUEA"
 80 PRINT"LETTRES : "
 90 PAPER1:FOR I=1 TO L
 100 CURSOR 2*I+8,0:PRINT CHR$(INT(RND(65,91)))
:BEEP
 110 IF RND(0,1)<.4 THEN CURSOR 2*I+8,0:PRINTMI
D$(A$,INT(RND(1,8)),1)
 120 NEXT I:PAPER0
 130 I=1
 140 CURSOR 0,3:PRINT"JOUEUR";I;:INPUT" LONGUEU
R :";L
 150 CURSOR 21,3:PRINT"      "
 160 IF I=1 THEN P=L
 170 I=I+1
 180 IF I<3 THEN GOTO 140
 190 S=S+(P AND P>L)
 200 J=J+(L AND L>P)
 210 CURSOR 0,6:PRINT"SCORES : JOUEUR 1 :";S:PR
INT"           JOUEUR 2 :";J
 220 CURSOR 0,10:PRINT"'ENTREE' POUR CONTINUER
":INPUT"'F' POUR TERMINER       ";R$
 230 IF R$<>"F" THEN GOTO 50
 240 STANDARD
```

Machinamot

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de hasard (sur des mots).
Difficulté du jeu : aucune.

LE JEU :
Ce très court programme permet à HECTOR de faire la démonstration de ses capacités à manipuler des chaînes de caractères : on part simplement d'un préfixe et d'un suffixe d'origine grecque, que l'on va assembler au hasard, on tombera bien sûr, de temps à autre, sur des mots absurdes ou même légèrement équivoques.
Un nouveau mot est confectionné à chaque essai, mais il est possible d'obtenir plusieurs fois le même mot.

LE PROGRAMME :
Deux variables A$ et B$ sont dimensionnées à 15 en ligne 5Ø et 6Ø : l'une contiendra les préfixes, l'autre les suffixes (si l'on désire rajouter des possibilités, il suffira de modifier cette limitation à 15).
La boucle I (7Ø-9Ø) lit les lignes DATA (16Ø-18Ø) concernant les préfixes suivis des suffixes.
La ligne 1ØØ effectue le choix d'un préfixe au hasard.
La ligne 11Ø effectue le choix d'un suffixe au hasard.
Le mot obtenu en ligne 12Ø est affiché par la ligne 13Ø, puis on recommence (ligne 15Ø).

POUR JOUER :
Lancer l'exécution du programme par RUN. Un premier mot sera immédiatement affiché, puis un nouveau mot sera affiché à la suite du précédent.
Taper CNT et C pour sortir du programme.

EXTENSIONS POSSIBLES :
On pourra très facilement augmenter la liste des mots disponibles.
On pourra aussi utiliser un procédé analogue pour faire confectionner des phrases à l'ordinateur, lui faire inventer des poèmes, etc. On pourra aussi transformer le programme initial en un jeu de société qui consisterait, par exemple, à prendre des paris sur le prochain mot qui va apparaître. Et confier, éventuellement, la gestion des paris à HECTOR...

```
10 ' MACHINAMOT
20 SEEDTIME(0)
40 CLS
50 DIMA$(15)
60 DIMB$(15)
70 FOR I=1 TO 15
80 READ A$(I),B$(I)
90 NEXT I
100 I=INT(RND(1,16))
110 J=INT(RND(1,16))
120 C$=A$(I)+B$(J)
130 PRINTC$:BEEP
140 PAUSE .6
150 GOTO 100
160 DATAANDRO,CRATE,COSMO,PHILE,MICRO,PHOBE,HE
LIO,TROPE,SEXO,GYNE,STEREO,MANE
170 DATAPALEO,NAUTE,MACRO,PHAGE,HYDRO,FUGE,STE
NO,PATHE,PATHO,CARDE,OLEO,DERME
180 DATAMYTHO,SCOPE,HOMEO,VORE,OCTO,PHONE
```

Accrochez les wagons

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : solitaire.
Difficulté du jeu : facile.

LE JEU :
Il faut essayer d'accrocher, le plus rapidement possible, la locomotive à son wagon, qui apparaissent tous deux en début de partie, en deux endroits plus ou moins éloignés de l'écran. On devra pour cela déplacer la locomotive en indiquant la suite des mouvements à effectuer (sous forme codée : H pour une case vers le haut ; B pour une case vers le bas ; D pour une case vers la droite et G pour une case vers la gauche).
On tapera, par exemple : BBGDHHH.

LE PROGRAMME :
Les lignes 4Ø et 5Ø dessinent la locomotive, la ligne 6Ø le wagon, et les roues sont en ligne 7Ø.
La position initiale de la locomotive est définie par les lignes 9Ø et 1ØØ, celle du wagon par les lignes 11Ø et 12Ø.
La ligne 13Ø évite qu'ils soient initialement trop proches l'un de l'autre.
Les lignes 15Ø et 16Ø réalisent l'affichage du wagon et les lignes 17Ø, 18Ø et 19Ø celui de la locomotive.
Le test de fin est en ligne 2ØØ.
Les mouvements à effectuer sont entrés dans une variable alphanumérique R$, en ligne 21Ø, et sont décodés dans une boucle I (24Ø-27Ø) pour être exécutés en ligne 28Ø.

POUR JOUER :
RUN permet d'afficher les positions de départ de la locomotive et du wagon.
On tape alors la série des déplacements à effectuer, sous la forme DDDHHHBHGG, suivie de ENTREE.
La locomotive sera alors affichée dans sa nouvelle position, et l'on recommence, jusqu'à ce qu'elle soit accrochée au wagon.
On obtiendra alors l'affichage du score.

EXTENSIONS POSSIBLES :
Il sera intéressant de prévoir la perte de la partie si, par accident, au cours du déplacement, la locomotive vient en collision avec le wagon.
On pourra aussi prévoir plusieurs wagons et un ordre d'accrochage, jouer à plusieurs avec plusieurs locomotives, dessiner les voies ferrées, introduire des obstacles à contourner obligatoirement d'un certain côté, etc.

```
 10 'ACCROCHEZ LES WAGONS
 20 CLS
 30 SEEDTIME(0)
 40 D$="   *"
 50  A$=CHR$(127) +CHR$(127)+CHR$(127)
 60 B$="(_)"
 70 C$="O O"
 80 N=0
 90 LY=INT(RND(0,31))
 100 LX=INT(RND(1,17))
 110 WY=INT(RND(0,31))
 120 WX=INT(RND(1,17))
 130 IF ABS(LX-WX)<4 THEN GOTO 90
 140 CLS
 150 CURSOR WY*6+9,226-WX*10:PRINTB$
 160 CURSOR WY*6+9,216-WX*10:PRINTC$
 170 CURSOR LY*6+9,236-LX*10:PRINTD$
 180 CURSOR LY*6+9,226-LX*10:PRINTA$

 190 CURSOR LY*6+9,216-LX*10:PRINTC$
 200 IF LX=WX AND WY=LY-3 THEN GOTO 300
 210 CURSOR 15,46:INPUT"CHEMIN A SUIVRE (DDGBHH
...) :";R$
 220 X=0:Y=0
 230 N=N+1
 240 FOR I=1 TO LEN(R$):X$=MID$(R$,I,1)
 250 X=X-(X$="B")+(X$="H")
 260 Y=Y-(X$="D")+(X$="G")
 270 NEXT I
 280 LX=LX+X:LY=LY+Y
 290 GOTO140
 300 CURSOR 15,46:PRINT"BRAVO ... TROUVE EN";N;
"COUPS."
 310 LASER
 320 END
```

Tour de cartes

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de prestidigitation.
Difficulté du jeu : facile.

LE JEU :
C'est un tour assez facile mais, ici, les cartes sont manipulées sous vos yeux par HECTOR qui tient le rôle du prestidigitateur ; 21 cartes, prises au hasard dans un paquet de 52, sont affichées, et le joueur devra retenir mentalement l'une des cartes présentées. Puis, il devra indiquer à HECTOR dans quelle colonne (1re, 2e ou 3e à partir de la gauche) se trouve la carte choisie ; les cartes sont alors redistribuées et le joueur indique de nouveau la colonne correspondant à sa carte ; ceci est fait encore une fois et, au 4e affichage, HECTOR est en mesure de deviner la carte choisie.

LE PROGRAMME :
La boucle I (60-120) permet de choisir, au hasard, 21 cartes sur 52, et la boucle emboîtée J (80-100) permet de s'assurer que l'on ne réalise pas deux fois le même tirage. La ligne 140 contient les couleurs des cartes ainsi que les noms des cartes figures (VA, DA, RO, AS), dans une variable alphanumérique B$.
La boucle I (150-240) réalise l'affichage du jeu.
La ligne 160 décode la couleur de la carte.
La ligne 170 choisit la position d'affichage.
La ligne 180 décode la hauteur.
L'affichage proprement dit est effectué de la ligne 190 à la ligne 230.
La ligne 260 permet au joueur d'entrer le numéro de la colonne où se trouve sa carte. La boucle J (300-320) réalise le brouillage, et les boucles emboîtées J (350-390) et I (360-380) fabriquent la nouvelle variable A$ à utiliser pour le prochain affichage.

POUR JOUER :
On lance le programme par RUN.
On obtient un premier affichage de 21 cartes réparties sur 3 colonnes.
On indique alors le numéro (1, 2 ou 3) de la colonne où se trouve la carte choisie, suivi de ENTREE.
On obtient un 2e affichage et l'on fait de même ; puis un 3e affichage et l'on fait de même.
Au 4e affichage, la carte choisie est automatiquement affichée.

EXTENSIONS POSSIBLES :
On peut envisager une amélioration de l'affichage en écrivant les noms des cartes en entier, ou même, en dessinant de véritables cartes, en couleur.
On pourra aussi essayer de programmer d'autres tours, utilisant le même principe de base, mais avec davantage de cartes, etc.

```
 10 'TOUR DE CARTES
 20 C$="TRCACOPI10VADAROAS"
 30 SEEDTIME(0)
 40 CLS
 50 A$=""
 60 FOR I=1 TO 21
 70 B$=CHR$(INT(RND(38,90)))
 80 FOR J=1 TO LEN(A$)
 90 IF B$=MID$(A$,J,1) THEN J=LEN(A$):NEXT J:GO
TO 70
 100 NEXT J
 110 A$=A$+B$
 120 NEXT I
 130 H=3:P=1
 140 B$=C$
 150 FOR I1=1 TO 7:FOR I2=0 TO 2:I=I2*7+I1
 160 S=INT((ASC(MID$(A$,I,1))-38)/13)
 170 J=7*INT((I-1)/7)
 180 T=ASC(MID$(A$,I,1))-37-13*S
 190  IF P=0 AND I<>11  THEN GOTO 240
 195  IF P=0 THEN PRINT "VOTRE CARTE EST :";
 210 IF T<=8 THEN PRINTSTR$(T+1);
 220 IF T>8 THEN PRINTMID$(B$,ABS(2*T-9),2);
 230 PRINT":";MID$(B$,2*S+1,2),
 240 NEXT I2,I1
 250 IF P=0 THEN GOTO 410
 260 PRINT:PRINT:PRINT:PRINT:INPUT"QUELLE COLON
NE ... :";N
 270 CLS
 280 H=H-1
 290 R$="20120":B$=""
 300 FOR J=0 TO 2
 310 B$=B$+MID$(A$,7*VAL(MID$(R$,N+J,1))+1,7)
 320 NEXT J
 330 IF H=0 THEN A$=B$ :P=0 :GOTO 140
 340 A$=""
 350 FOR J=0 TO 2
 360 FOR I=1 TO 7
 370 A$=A$+MID$(B$,3*I-2+J,1)
 380 NEXT I
 390 NEXT J
 400 GOTO 140
 410 END
```

L'indésirable

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : hasard et réflexion (1 joueur).
Difficulté du jeu : moyenne.

LE JEU :
Vous êtes le propriétaire d'un immeuble de 16 pièces vides, et vous n'arrivez pas à mettre la main sur votre locataire indésirable. Vous visitez les pièces au hasard, mais, si vous visitez la pièce voisine, votre locataire, prévenu par le bruit, va déménager. Il peut même se réinstaller dans une pièce que vous avez déjà visitée, et marquée comme vide. Il vous faudra arriver à le déloger en moins de 9 coups pour gagner.

LE PROGRAMME :
Une table B est dimensionnée à 16, en ligne 60, pour recevoir les numéros des cases déjà visitées.
La boucle I (80-110) dessine le terrain de jeu.
La position initiale du locataire est tirée au hasard en ligne 120.
Le numéro de la case à visiter est donné en ligne 130.
Les lignes 140 à 220 contrôlent les diverses réponses possibles.
La boucle I (180-200) dessine les cases visitées.
La ligne 230 dessine le locataire, effacé en ligne 260.
Les lignes 270 et 280 assurent son déplacement.

POUR JOUER :
On lance l'exécution par RUN.
A la question CASE VISITEE, on répond en tapant une lettre, de A à P. Si on a trouvé le locataire, celui-ci est affiché, et la partie est terminée ; sinon, deux cas peuvent se produire : soit l'on est suffisamment loin, et le locataire ne bouge pas, soit l'on est à une case, il apparaît un court instant, et il se recache, aléatoirement, dans une case voisine. Les cases déjà visitées se colorent en rouge, et vous ne pouvez pas y revenir. Si, au bout de 9 coups, vous n'avez pas trouvé, la position du locataire s'affiche, et vous avez perdu.

EXTENSIONS POSSIBLES :
On pourra, bien sûr, jouer sur plus de 16 cases, utiliser des caractères graphiques pour représenter le locataire et le propriétaire, utiliser un algorithme plus subtil pour le déplacement du locataire, lui interdire éventuellement de revenir dans des cases déjà visitées, etc.

```
 10 'L'INDESIRABLE
 20 CLS:SPECIAL:SCALE1,1
 40 A$="#
 50 B$=A$+A$+A$
 60 DIMB(16)
 70 CLS
 80 PEN 1:FOR I=1 TO 16
 90 CURSOR 4+4*(I-1-4*INT((I-1)/4)),2+4*INT((I-
1)/4):PRINTB$
 100 CURSOR 4+4*(I-1-4*INT((I-1)/4)),3+4*INT((I
-1)/4):PRINTA$;:PEN2:PRINTCHR$(64+I);:PEN1:PRIN
TA$
 105 CURSOR 4+4*(I-1-4*INT((I-1)/4)),4+4*INT((I
-1)/4):PRINTB$
 110 NEXT I
 120 X=INT(RND(1,17)):S=1:K=0
 130 CURSOR 0,22:PEN 2:INPUT"CASE VISITEE :";C$
:C=ASC(C$)-64
 140 IF B(C)=1 THEN BEEP:PRINT:PRINT"DEJA VISIT
EE ...":PAUSE 1:CURSOR 0,24:PRINT"
      ":GOTO 130
 150 IF C=X THEN CURSOR 0,24:LASER:PRINT"VOUS G
AGNEZ EN";S;"COUPS.":K=1:GOTO 230
 160 IF X-5=C OR X-4=C OR X-3=C OR X-1=C OR X+1
=C OR X+3=C OR X+4=C OR X+5=C THEN GOTO 230
 170 CURSOR 0,24:PRINT"IL N'EST PAS LA...":SHOT
 1
 180 S=S+1:B(C)=1:FOR I=1 TO 16
 190 CURSOR 4+4*(I-1-4*INT((I-1)/4)),2+4*INT((I
-1)/4):PEN 1+2*B(I):PRINTB$
 200 NEXT I
 210 IF S=9 THEN SIREN:CURSOR 0,24:PRINT"VOUS A
VEZ PERDU":PRINT:PRINT"IL ETAIT EN ";CHR$(X+64)
:K=1:GOTO 230
 220 CURSOR 0,24:PRINT"
  ":GOTO 130
 230 CURSOR 5+4*(X-1-4*INT((X-1)/4)),1+4*INT((X
-1)/4):PEN 3:PRINT"%"
 235 IF K=0 THEN SHOT 0
 240 PAUSE .5
 250 IF K=1 THEN PAUSE 2:HUSH:STANDARD:PEN 2:EN
D
 260 CURSOR 5+4*(X-1-4*INT((X-1)/4)),1+4*INT((X
-1)/4):PRINT" "
 270 Z=INT(RND(1,11))-5:IF ABS(Z)=2 THEN GOTO 2
70
 280 X=X+Z:IF X>16 OR X<1 THEN X=X-Z:GOTO 270
 290 GOTO 180
```

Le jeu de la vie

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu solitaire (2 versions).
Difficulté du jeu : facile.

LE JEU :
La version présentée ici respecte les règles posées à l'origine par le mathématicien J.H. CONWAY.
Une certaine population est définie au début du jeu soit par HECTOR (version automatique) soit par vous-même (version manuelle). Cette population va évoluer sous vos yeux, en respectant certaines règles, très élémentaires de naissance, de vie et de mort. L'évolution se poursuivra jusqu'à ce que l'on décide d'arrêter le programme.

LE PROGRAMME :
Deux tables (B et C) sont utilisées en ligne 5Ø, pour enregistrer l'état de la population. Les boucles F (23Ø-31Ø) et G (24Ø-3ØØ) sont utilisées en début de jeu pour définir la population initiale (lignes 25Ø et 26Ø). Le calcul de la nouvelle population est ensuite réalisé à l'aide d'une quadruple boucle : 8Ø-21Ø, 9Ø-2ØØ, 11Ø-16Ø, 12Ø-15Ø. La ligne 13Ø réalise la gestion du nouvel état suivant les règles.
L'affichage de ce nouvel état est obtenu en réutilisant la double boucle d'affichage F (23Ø-31Ø) et G (24Ø-3ØØ).
La ligne 32Ø permet de recommencer.

POUR JOUER :
Dans la version automatique, il suffit de lancer le programme par RUN et de regarder. Pour arrêter, il faudra taper CNT et C. Dans la version manuelle, la configuration de départ sera entrée, après le lancement du programme par RUN, en tapant Ø si l'on ne veut rien mettre sur la ligne et la colonne indiquée, et 1 si l'on désire y mettre une cellule. Puis l'évolution se déroulera ensuite de façon automatique.

EXTENSIONS POSSIBLES :
Il sera facile, si l'on dispose d'une mémoire plus importante, d'augmenter les dimensions des tables B et C, en redéfinissant la variable A. On pourra aussi introduire la couleur, améliorer la précision du dessin en remplaçant les PRINT par un affichage haute résolution.

```
 10 'LE JEU DE LA VIE (AUTOMATIQUE)
 20 CLS:STANDARD:COVER 1
 30 SEEDTIME(0)
 40 INPUT"DIMENSION ... :";A:CLS
 50 DIM B(A,A),C(A,A)
 60 I=0
 70 GOTO 220
 80 FOR F=1 TO A
 90 FOR G=1 TO A
 100 S=0
 110 FOR I=-1 TO 1
 120 FOR J=-1 TO 1
 130 IF F+J=0 OR G+I=0 OR F+J=A+1 OR G+I=A+1 TH
EN GOTO 150
 140 S=S+B(F+J,G+I)
 150 NEXT J
 160 NEXT I
 170 S=2*S-B(F,G)
 180 C(F,G)=0
 190 IF S>=5 AND S<=7 THEN C(F,G)=1
 200 NEXT G
 210 NEXT F
 220 HOME
 230 FOR F=1 TO A
 240 FOR G=1 TO A
 245 BEEP
 250 IF I<>0 THEN GOTO 270
 260 IF RND(0,1)<.5 THEN C(F,G)=1
 270 IF C(F,G)1 THEN PEN 1:PRINT" *";:ELSE PEN
2:PRINT" .";
 280 B(F,G)=C(F,G)
 290 NEXT G
 300 PRINT:PRINT
 310 NEXT F
 320 PEN 2:GOTO 80
```

```
 10 'LE JEU DE LA VIE (MANUEL)
 20 CLS:SPECIAL:SCALE 1,1
 30 SEEDTIME(0)
 40 INPUT"DIMENSION ... :";A:CLS
 50 DIM B(A,A),C(A,A)
 60 I=0
 70 GOTO 220
 80 FOR F=1 TO A
 90 FOR G=1 TO A
 100 S=0
 110 FOR I=-1 TO 1
 120 FOR J=-1 TO 1
 130 IF F+J=0 OR G+I=0 OR F+J=A+1 OR G+I=A+1 TH
EN GOTO 150
 140 S=S+B(F+J,G+I)
 150 NEXT J
 160 NEXT I
 170 S=2*S-B(F,G)
 180 C(F,G)=0
 190 IF S>=5 AND S<=7 THEN C(F,G)=1
 200 NEXT G
 210 NEXT F
 220 CLS
 230 FOR F=1 TO A
 240 FOR G=1 TO A
 245 BEEP
 250 IF I<>0 THEN GOTO 280
 260 PEN 2:CURSOR 0,23:PRINT"LIGNE";F;"COLONNE"
;G:PRINT:PRINT"TAPER 1 OU 0 ..."
 270 CURSOR 2*G,2*F:INPUT"";C(F,G)
 280 CURSOR 2*G,2*F:IF C(F,G)1 THEN PEN 1:PRINT
" *";:ELSE PEN 2:PRINT" .";
 290 B(F,G)=C(F,G)
 300 NEXT G
 310 NEXT F
 320 PEN 2:GOTO 80
```

Niveau 3

Nom du programme	*Thèmes et particularités*
Un dé	Tirage d'un dé, avec graphisme.
Plus grand, plus petit	Un jeu de déduction intéressant, pour un joueur.
Ordre	Un peu d'astuce pour remettre en ordre, le plus vite possible.
Le pendu	Un jeu de lettres très classique (2 joueurs).
Belote	Pour simuler une donne, avec un jeu de 32 cartes.
Un taquin de calcul	Un bon entraînement (2 joueurs).
Pendu inversé	Cette fois, c'est l'ordinateur qui essaie de deviner le mot.
Nicomaque	Pour se perfectionner en calcul mental (1 ou plusieurs joueurs).
Course de chevaux	Assez visuel : un jeu de hasard, pour plusieurs joueurs.
A travers	Un jeu de stratégie pour 2 joueurs.
Mastermind	Avec des chiffres, mais très classique. Un seul joueur.
Idem	Un jeu de calcul mental, avec plusieurs niveaux, pour plusieurs joueurs.
Horloge 1	Ne jetez pas votre vieille télévision : transformez-la en horloge.
Reines sur l'échiquier	Un vieux problème, pour vous aider à réfléchir.
Répétitions	Retrouver une séquence de chiffres de plus en plus longue.
Mots	Une idée très simple, mais ce n'est pas si facile de trouver !
Permutations	Un jeu de réflexion, pas toujours évident.
Bridge	Mettez au point vos propres problèmes de bridge.
Hanoï	Un conseil : commencez petit ! ! !
Le loup et les agneaux	Un jeu de réflexion pour 2 joueurs.

Manipulation des chaînes

TRAITEMENT DES CHAINES DE CARACTERES : MID$, LEFT$, RIGHT$

Nous avons eu l'occasion de rencontrer, dès le niveau II, la fonction essentielle utilisée dans le Basic B3X pour manipuler les chaînes : il s'agit de MID$ (A$, I J) qui permet d'extraire, dans la chaîne A$, une sous-chaîne de longueur J, commençant au Ième caractère.

Ainsi si,

A$ = "MONSIEUR"

on aura :

MID$ (A$, 2, 3) = "ONS"

Pour faciliter les manipulations, il est possible d'utiliser les deux fonctions LEFT$ et RIGHT$.

LEFT$ (A$, I) permet d'extraire les I premiers caractères à partir de la gauche du mot A$.

Ainsi, LEFT$ (A$, 2) = "MO", avec l'exemple précédent (on pourrait d'ailleurs utiliser MID$ (A$, 1, 2) pour obtenir le même résultat).

RIGHT$ (A$, J) permet d'extraire les J derniers caractères (on part de la droite du mot A$).

Ainsi RIGHT$ (A$, 3) = "EUR", toujours avec le même exemple.

Il est plus difficile de se passer de la fonction RIGHT$, mais il serait possible de faire MID$ (A$, LEN (A$) - 3, 3) pour obtenir le même résultat.

Le Basic de HECTOR ne permet pas le remplacement direct d'une sous-chaîne par une affectation du type MID$ (-.-) = "   ". Il sera nécessaire, pour remplacer la Ième lettre de la chaîne A$, d'écrire :

A$ = LEFT$ (A$, I – 1) + "nouvelle lettre" + MID$ (A$, I + 1).

Des formules analogues pourront être utilisées pour remplacer dans A$ une sous-chaîne par une autre, pas nécessairement de même longueur, ce qui permet une grande souplesse.

Un dé

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : tirage d'un dé (avec graphisme).
Difficulté du jeu : aucune.

LE JEU :
Le programme propose simplement le lancer d'un dé aléatoire, et sa représentation graphique en grand format.
Il pourra facilement être utilisé en sous-programme, dans tout type de jeux faisant appel au lancer d'un ou de plusieurs dés.

LE PROGRAMME :
La ligne 5Ø utilise le générateur aléatoire de HECTOR pour tirer un nombre de 1 à 6.
La boucle J (7Ø-9Ø) dessine les bords du dé, et les lignes 6Ø et 1ØØ dessinent le haut et le bas.
Les lignes 12Ø à 17Ø sont utilisées pour le dessin des points.
Les lignes 18Ø à 2ØØ permettront de refaire un tirage.

POUR JOUER :
Lancer le programme par RUN.
Un dé s'affiche, en couleurs, et en grand format.
Après une pause de 2 secondes, un nouveau dé s'affiche. Faire CNT et C pour arrêter.

EXTENSIONS POSSIBLES :
On pourra introduire une petite musique agrémentant chaque lancer, faire défiler les points comme si le dé roulait sur une table, etc.

```
10 'UN DE
20 CLS:SPECIAL:SCALE1,2
30 SEEDTIME(0)
40 CLS
50 A=INT(RND(1,7))
60 PAPER 1:CURSOR 7,0:PRINT"             "
70 FOR J=1 TO 11
80 CURSOR 7,J:PRINT" ":CURSOR 19,J:PRINT" "
90 NEXT J
100 CURSOR 7,12:PRINT"             "
110 PAPER3
120 IF A<>1 THEN CURSOR 10,3:PRINT" "
130 IF A>=4 THEN CURSOR 16,3:PRINT" "
140 IF A=6 THEN CURSOR 10,6:PRINT" ":CURSOR 16
,6:PRINT" "
150 IF A/2<>INT(A/2) THEN CURSOR 13,6:PRINT" "

160 IF A>=4 THEN CURSOR 10,9:PRINT" "
170 IF A<>1 THEN CURSOR 16,9:PRINT" "
180 PAUSE2
190 PAPER0
200 GOTO40
```

Plus grand, plus petit

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de déduction (un seul joueur).
Difficulté du jeu : difficile.

LE JEU :
Il s'inspire d'un principe légèrement différent de celui du Mastermind ; il s'agit également de deviner un nombre (de 1 à 4 chiffres), mais les indications dont on dispose sont cette fois d'une part le nombre de chiffres qui, dans le nombre essayé, sont supérieurs à ceux du nombre à deviner, et, d'autre part, le nombre de chiffres qui sont inférieurs. Si, par exemple, le nombre à deviner est 3Ø54, et que l'on essaie 4251, la machine répondra 2 plus grands (ce sont le 4 et le 2), et 1 plus petit (c'est le 1).

LE PROGRAMME :
La ligne 6Ø interdit de demander des nombres de plus de 4 chiffres.
Le nombre à deviner est placé en ligne 8Ø, dans une variable alphanumérique X$.
Les comparaisons avec le nombre essayé sont faites à la ligne 13Ø (qui teste si l'on a trouvé juste), et dans la boucle I (15Ø-18Ø) où l'on fait la comparaison "lettre à lettre".
L'affichage du résultat est réalisé grâce à la ligne 19Ø.
Le compteur du nombre d'essais est en ligne 2ØØ.

POUR JOUER :
Le lancement du programme par RUN provoque l'affichage NOMBRE DE CHIFFRES.
On entre un nombre inférieur à 5 et on appuie sur la touche ENTREE.
Le numéro de l'essai s'affiche, ainsi que la question NOMBRE ESSAYE ; on répond en entrant un nombre, suivi de ENTREE.
Le résultat est alors affiché sous la forme > et <.
S'il n'est pas exact, on recommence (attention à entrer uniquement des nombres ayant autant de chiffres que le nombre à deviner).
Si l'on a trouvé HECTOR affichera TROUVE... EN... COUPS. Puis AUTRE PARTIE (O/N) ; on fera O suivi de ENTREE pour commencer une nouvelle partie.

EXTENSIONS POSSIBLES :
On pourra, comme pour le Mastermind, imaginer un jeu analogue avec des lettres, des couleurs, ou même en utilisant d'autres symbolismes.
On peut aussi prévoir de laisser jouer HECTOR qui tiendra alors le rôle de décodeur.

```
10 'PLUS GRAND, PLUS PETIT
20 CLS:SPECIAL:SCALE1,2
30 SEEDTIME(0)
40 CLS
50 INPUT"NB. DE CHIFFRES :";C
60 IF C>5 THEN GOTO 40
70 CLS:E=1:F=1
80 X$=RIGHT$(STR$(100000+INT(RND(0,10^C+1))),C
)
100 CURSOR 0,0:PRINT"ESSAI N.  NB.ESSAYE  >  <
"
110 CURSOR 0,1:PRINT"______________________________
"
120 CURSOR 0,E+2:PRINTTAB(4);F;TAB(10);:INPUT"
";P$
130 IF P$=X$ THEN GOTO 220
140 A=0:B=0
150 FOR I=1 TO C
160 IF MID$(P$,I,1)<MID$(X$,I,1) THEN A=A+1
170 IF MID$(P$,I,1)>MID$(X$,I,1) THEN B=B+1
180 NEXT I
190 CURSOR 20,E+2:PRINTA;B:BEEP
200 E=E+1:F=F+1:IF E>10 THEN E=1
210 GOTO 120
220 LASER:PRINT:PRINTX$;" TROUVE EN";F;"COUPS.
"
230 CURSOR 0,12:INPUT"AUTRE PARTIE (O/N) :";R$

240 IF R$="O" THEN GOTO 40
250 STANDARD
```

Ordre

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexion.
Difficulté du jeu : moyenne.

LE JEU :
Les 1Ø premières lettres de l'alphabet sont affichées dans le désordre ; il faut réussir à les remettre dans l'ordre, avec le moins de coups possible, par des permutations successives de deux des lettres ; mais ceci a aussi pour inconvénient de faire permuter autour d'elles les deux lettres qui entourent celles-ci, suivant le mécanisme expliqué ci-dessous :

```
B A D G H I J C F E
Ø 1 2 3 4 5 6 7 8 9   : permutation 83

B A H F D J I E G C
Ø 1 2 3 4 5 6 7 8 9   : résultat
```

LE PROGRAMME :
Les boucles emboîtées I (6Ø-12Ø) et J (8Ø-1ØØ) définissent une variable alphanumérique A$, contenant les 1Ø premières lettres de l'alphabet dans le désordre.
La ligne 14Ø affiche A$, et la ligne 15Ø affiche les numéros de lettres de l'alphabet pour la permutation.
On entre la permutation en ligne 19Ø.
Les lignes 2ØØ et 21Ø transforment la variable alphanumérique entrée en une variable numérique.
La ligne 22Ø appelle une première fois le sous-programme d'inversion.
La ligne 25Ø appelle une deuxième fois le sous-programme d'inversion.
La ligne 26Ø comptabilise les essais.
La ligne 18Ø contrôle les résultats.

POUR JOUER :
Lancer le programme par RUN.
Le mot initial est affiché, chaque lettre étant repérée par un numéro de 1 à 9.
A la question INVERSION (I/J), on répond en entrant un nombre de deux chiffres xy, où x est le numéro de la première lettre, et y le numéro de la seconde. Les deux inversions sont alors successivement effectuées, et l'on recommence.
Le nombre de coups sera affiché en fin de partie.

EXTENSIONS POSSIBLES :
On peut imaginer un jeu contre l'ordinateur, qui évaluerait le score du joueur en fonction de ses propres performances ; on peut aussi modifier les règles qui régissent les permutations.

```
 10 'ORDRE
 20 SEEDTIME(0)
 30 CLS:SPECIAL:SCALE 1,2:PEN2
 50 A$=""
 60 FOR I=1 TO 10
 70 R$=CHR$(INT(RND(65,75)))
 80 FOR J=1 TO I-1
 90 IF R$=MID$(A$,J,1) THEN J=I:NEXT J:GOTO 70
 100 NEXT J
 110 A$=A$+R$
 120 NEXT I
 130 S=1
 140 CURSOR 0,0:PEN2:PRINT"RESULTAT : ";:PEN 1:
PRINTA$
 150 PEN 3:PRINTTAB(12);"0123456789"
 160 PEN 2:CURSOR 0,5:PRINT"ESSAI NUMERO";S
 170 CURSOR 21,8:PRINT"          "
 180 IF A$="ABCDEFGHIJ" THEN GOTO 280
 190 CURSOR 0,8:INPUT"INVERSION (FORME IJ)";R$
 200 I=ASC(LEFT$(R$,1))-47
 210 J=ASC(RIGHT$(R$,1))-47
 220 GOSUB 500
 230 IF LEFT$(R$,1)="0" OR LEFT$(R$,1)="9" THEN
 GOTO 250
 240 I=I-1:J=I+2
 250 GOSUB 500
 260 S=S+1
 270 GOTO 140
 280 PRINT:PRINT"TROUVE EN";S-1;"COUPS."
 290 STANDARD
 300 END
 500 B$=""
 510 FOR T=1 TO LEN(A$)
 520 IF T=I THEN N$=MID$(A$,T,1)
 530 IF T=J THEN M$=MID$(A$,T,1)
 540 NEXT T
 550 FOR T=1 TO LEN(A$)
 560 IF T=I THEN B$=B$+M$:GOTO 590
 570 IF T=J THEN B$=B$+N$:GOTO 590
 580 B$=B$+MID$(A$,T,1)
 590 NEXT T
 600 A$=B$
 610 RETURN
```

Le pendu

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de lettres (version deux joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Il serait sans doute assez difficile de retrouver le nom de l'inventeur de ce jeu très à la mode, mais il est certain qu'il existe depuis des temps immémoriaux, quelquefois pratiqué sous des formes assez différentes (par exemple, reconstituer un petit cochonnet en lançant des dés).
L'apparition des jeux sur micro-ordinateur et la magie des écrans de télévision lui ont fait connaître un renouveau certain : dans la version retenue, l'un des joueurs doit entrer un mot de longueur quelconque. L'ordinateur affichera alors des tirets à l'emplacement des lettres de ce mot et demande au deuxième joueur de proposer une lettre. Chaque réponse correcte fait apparaître la lettre à son emplacement exact dans le mot, et chaque réponse fausse fait progresser le dessin de la potence.
Pour gagner, il suffit de réussir à trouver le mot inconnu avant d'être pendu.

LE PROGRAMME :
Le mot à deviner est introduit en début de programme (ligne 7Ø) ; une variable B$, en ligne 25, contient tous les éléments du dessin.
La ligne 1ØØ fabrique un mot P$ qui contient des tirets à l'emplacement de chacune des lettres du mot initial.
Le sous-programme 28Ø-32Ø s'occupe du dessin (plus ou moins complet, selon la valeur de M).
Les lignes 19Ø-2ØØ affichent le résultat et arrêtent le jeu.
La lettre essayée est entrée en ligne 12Ø et testée dans une boucle I (14Ø-16Ø) ; le sous-programme 22Ø-24Ø remplace chaque tiret par la lettre correspondante.

POUR JOUER :
Après le lancement du programme par RUN, l'un des joueurs doit entrer son mot et taper ENTREE. Des tirets sont affichés à l'emplacement de chacune des lettres du mot ; et la question VOTRE LETTRE apparaît sur l'écran ; l'autre joueur répond alors en proposant une lettre suivie de ENTREE.
Si la lettre est bonne, elle viendra remplacer un ou plusieurs tirets ; sinon, la potence va commencer à se dessiner.
Le mot exact sera affiché dans le cas où l'on aurait perdu.

EXTENSIONS POSSIBLES :
Il sera intéressant de constituer une liste de mots dans lesquels HECTOR pourra piocher au hasard ; on pourra aussi améliorer le dessin et modifier le nombre d'essais permis selon la difficulté du mot.

```
 10 'LE PENDU
 20 CLS:SPECIAL:SCALE1,1:PEN2
 25 B$="###### #   ! ############ .- !   !. ! -
!  !  !   !  !  !  !!  !  !  !  "
 30 DATA3,5,7,5,7,2,7,8,11,3,11,7
 40 FOR I=1 TO 6:READ V(I),H(I):NEXT I
 50 CLS
 60 FOR I=1 TO 15:CURSOR 1,I:PRINT"..........."
:NEXT
 70 CURSOR 12,3:PRINT"VOTRE MOT :":CURSOR 12,5:
INPUT"==>";R$
 80 L=LEN(R$):IF L>14 OR L=0 THEN BELL:GOTO 70
 90 CURSOR 12,3:PRINT" DE";L;"LETTRES"
 100 P$=LEFT$("--------------",L)
 110 CURSOR 12,5:PRINTP$;"     "
 120 CURSOR 12,9:PRINT"VOTRE LETTRE :";:K$=INST
R$(1):PRINTK$
 130 T=0
 140 FOR I=1 TO L
 150 IF MID$(R$,I,1)=K$ THEN GOSUB 220
 160 NEXT I
 170 IF T=0 THEN M=M+1:GOSUB 280
 180 CURSOR 12,5:PRINTP$
 190 IF P$=R$ THEN CURSOR 0,25:LASER:PRINT"VOUS
 AVEZ TROUVE ...":STANDARD:END
 200 IF M=6 THEN CURSOR 0,25:LASER:PRINT"DOMMAG
E PERDU : ";R$:STANDARD:END
 210 GOTO 120
 220 T=1
 230 P$=LEFT$(P$,I-1)+K$+MID$(P$,I+1)
 240 RETURN
 270 END
 280 PEN 1:CURSOR H(M),V(M):PRINTMID$(B$,M*12-1
1,3)
 290 CURSOR H(M),V(M)+1:PRINTMID$(B$,M*12-8,3)
 300 CURSOR H(M),V(M)+2:PRINTMID$(B$,M*12-5,3)
 310 CURSOR H(M),V(M)+3:PRINTMID$(B$,M*12-2,3)
 320 PEN 2:RETURN
```

Belote

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de cartes.
Difficulté du jeu : facile.

LE JEU :
Le jeu proposé est plutôt une simulation, réalisée sur 32 cartes, de la distribution de 4 paquets de 5 cartes. Il pourra être utilisé pour l'étude de problèmes faisant intervenir des donnes de 2Ø cartes sur 32.

LE PROGRAMME :
En ligne 5Ø, on crée un tableau pouvant contenir les 2Ø cartes.
La boucle I (8Ø-14Ø) assure le tirage aléatoire de 2Ø nombres, pris parmi 32 ; la boucle J (11Ø-13Ø) vérifie que l'on ne réalise pas deux fois le même tirage.
La deuxième boucle I (16Ø-3ØØ) assure ensuite l'affichage, les noms des cartes étant affichés en clair, et la couleur étant décodée et affichée en clair, grâce aux lignes 17Ø à 27Ø.

POUR JOUER :
Il suffit de taper RUN pour obtenir immédiatement l'affichage de 4 paquets de 5 cartes chacun (une donne).
Un nouveau RUN permet d'obtenir une nouvelle distribution.

EXTENSIONS POSSIBLES :
On pourra facilement mettre au point un affichage plus graphique ; on pourra également se servir de ce programme comme d'un noyau pour la mise au point de programmes de jeux de cartes sur l'ordinateur (qui pourra, si nécessaire, servir de partenaire : voir l'exemple du Poker, dans le même ouvrage).

```
10 'BELOTE
20 SEEDTIME(0)
30 PAPER0:PEN2
40 STANDARD:CLS
50 DIM A(20)
60 K=1
70 PRINT"PATIENTEZ ! JE DISTRIBUE ..."
80 FOR I=1 TO 20
90 A(I)=INT(RND(1,33))
100 IF I=1 THEN GOTO 140
110 FOR J=1 TO I-1
120 IF A(I)=A(J) THEN J=I:NEXT J:GOTO 90
130 NEXT J
140 NEXT I
150 CLS 2:PRINT:PRINT:PRINT
160 FOR I=1 TO 20
170 S$="CA"
180 IF A(I)<=24 THEN S$="TR"
190 IF A(I)<=16 THEN S$="PI"
200 IF A(I)<=8 THEN S$="CO"
210 S=1+A(I)-8*INT(A(I)/8)
220 R$=RIGHT$(STR$(S+6),2)
230 IF S=5 THEN R$="VA"
240 IF S=6 THEN R$="DA"
250 IF S=7 THEN R$="RO"
260 IF S=8 THEN R$="AS"
265 PEN  1 AND (S$="CO" OR S$="CA")
270 PRINTR$;":";S$;"     ";
280 K=K+1:IF K=5 THEN K=1:PRINT:PRINT
290 BEEP
300 NEXT I
310 PAPER0:PEN2
```

Un taquin de calcul

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : difficile.

LE JEU :
Il se pratique à deux joueurs, l'un tenant le rôle du codeur, l'autre celui du décodeur.
Le codeur part d'un nombre (proposé par HECTOR) et doit le transformer à l'aide de 5 opérations qu'il choisit lui-même, ainsi d'ailleurs que les opérateurs.
Le résultat qu'il a obtenu et les opérations qu'il a utilisées seront affichés sur l'écran : le décodeur devra alors essayer de retrouver le nombre initial en effectuant les opérations en sens inverse.

LE PROGRAMME :
La ligne 5Ø permet d'obtenir l'entier de départ compris entre 5 et 2Ø.
Une boucle I (11Ø-2ØØ) permet d'entrer successivement les 5 opérations.
4 sous-programmes sont utilisés pour les calculs (aux lignes 1ØØØ et 4ØØØ).
Après chaque calcul, le résultat est affiché (seul le codeur devra regarder l'écran pendant cette opération) et le résultat définitif restera affiché pour le décodeur à la fin des calculs (ligne 19Ø).
La ligne 21Ø affiche les 5 opérations utilisées, ainsi que les opérateurs correspondants.

POUR JOUER :
On frappe RUN pour lancer le programme.
Le nombre du début est affiché, puis la question ENTREZ 5 OPERATIONS ; on devra les entrer l'une après l'autre, sous la forme, par exemple : + 4 suivi de ENTREE.
Le résultat sera affiché après chaque opération entrée. A la fin, les opérations entrées sont toutes affichées sur une même ligne, et le message BON COURAGE est envoyé à l'adresse du décodeur (il n'y a pas de contrôle au décodage).

EXTENSIONS POSSIBLES :
On pourra, bien sûr, songer à améliorer la partie décodage du programme en introduisant des contrôles sur le résultat ; on peut aussi envisager de laisser HECTOR participer davantage en lui confiant le rôle du codage, etc.

```
 10 'UN TAQUIN DE CALCUL
 20 SEEDTIME(0)
 30 PAPERO:PEN2
 40 SPECIAL:CLS:SCALE 1,2
 50 X=INT(RND(5,21)):BEEP
 60 PRINT"NOMBRE INTIAL :";X
 70 PRINT"ENTRER 5 OPERATIONS"
 80 PRINT"SOUS LA FORME +5 /2 *7 ...."
 90 F=X
 100 DIM R$(5)
 110 FOR I=1 TO 5
 120 CURSOR 0,5:INPUT"VOTRE OPERATION :";R$(I)
 130 CURSOR 0,5:PRINT"

 140 A=ASC(LEFT$(R$(I),1))-41
 150 IF A<=0 OR A=3 OR A=5 OR A>6 THEN BEEP:GOT
O 120
 160 IF A>3 THEN A=(A/2)+1
 170 ON A GOSUB 1000,2000,3000,4000
 180 CURSOR 14,7:PRINT"            "
 190 CURSOR 0,7:PRINT"NOMBRE OBTENU :";F
 200 NEXT I
 210 PRINT:PRINT"OPERATIONS";:FOR I=1 TO 5:PRIN
T" ";R$(I);:NEXT I
 220 PRINT:PRINT:PRINT"BON COURAGE ..."
 230 LASER
 240 STANDARD
 250 END
 1000 F=F*VAL(MID$(R$(I),2))
 1010 RETURN
 2000 F=F+VAL(MID$(R$(I),2))
 2010 RETURN
 3000 F=F-VAL(MID$(R$(I),2))
 3010 RETURN
 4000 F=F/VAL(MID$(R$(I),2))
 4010 RETURN
```

Pendu inversé

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu sur les mots.
Difficulté du jeu : facile.

LE JEU :
Le Pendu est un jeu de lettres très connu, présenté par ailleurs dans ce recueil, dans lequel il s'agit de deviner un mot caché par l'ordinateur et où chaque lettre non découverte vous rapproche un peu plus de la potence. Ici, c'est HECTOR qui va chercher à deviner votre mot, mais il joue quelque peu au hasard et ne possède pas votre intuition ; il faudra donc, en règle générale, beaucoup d'essais pour découvrir la solution. Pour augmenter ses chances, on a fait appel à une formule qui lui permet de tirer davantage dans les voyelles puis, dans les consonnes courantes, et, enfin, dans les consonnes plus rares.

LE PROGRAMME :
La ligne 5Ø permet d'enregistrer, dans une variable alphanumérique D$, les lettres de l'alphabet, dans l'ordre : voyelles, consonnes usuelles, autres consonnes. Le mot à deviner est introduit en ligne 7Ø (en fait, il suffirait d'introduire sa longueur).
La boucle I (9Ø-12Ø) imprime des tirets aux emplacements des lettres à trouver.
La boucle en ligne 14Ø sert uniquement à simuler le temps de recherche d'une lettre.
La formule, placée en ligne 15Ø, permet de modifier légèrement la probabilité de tirage, en faveur des voyelles et des consonnes usuelles.
La ligne 17Ø provoque un nouveau tirage, et la boucle I (21Ø-24Ø) réimprime les tirets et les lettres trouvées.
La ligne 25Ø incrémente le compteur d'essais.
La ligne 26Ø affiche le nombre d'essais réalisés, en fin de partie.

POUR JOUER :
Lancer le programme par RUN.
Entrer le mot à découvrir, suivi de ENTREE.
Des tirets sont alors affichés (un par lettre) et HECTOR propose une première lettre, regarde si elle convient et continue automatiquement, jusqu'à ce qu'il ait trouvé.
On pourra sortir du programme à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
Il sera sans doute intéressant, pour de jeunes enfants, de prévoir une version non automatique où le joueur répondrait OUI ou NON pour chaque lettre proposée par HECTOR ; on pourra aussi imaginer d'ajouter un affichage graphique, par exemple une potence ou une voiture qui avance vers un précipice, etc.

```
 10 'PENDU INVERSE
 20 SEEDTIME(0)
 30 PAPER0:PEN2
 40 CLS:SPECIAL:SCALE 1,2
 50 D$="YAEIOUBCDFGLMNPRSTVWHKJQXZ"
 60 C=1:T$=""
 70 INPUT"ENTREZ VOTRE MOT";R$
 80 CLS:CURSOR 0,11:PRINTD$
 90 FOR I=1 TO LEN(R$)
 100 CURSOR I,1:PRINT"-"
 110 T$=T$+"-"
 120 NEXT I
 130 CURSOR 0,3:PRINT"ESSAI N.";C:PRINT"
     JE CHERCHE ..."
 140 PAUSE 1
 150 X=-6*(RND(0,1)<.7)-13*(RND(0,1)<.5)-7*(RND
(0,1)<.4)
 160 X=INT(RND(1,X+1))
 170 IF MID$(D$,X,1)="-" THEN GOTO 150
 180 E$=MID$(D$,X,1)
 190 D$=LEFT$(D$,X-1)+"-"+MID$(D$,X+1):CURSOR 0
,11:PRINTD$
 195 SHOT1
 200 CURSOR 0,6:PRINT"JE PRENDS :";:PEN 1:PRINT
E$:PEN 2
 210 FOR I=1 TO LEN(R$)
 220 IF E$=MID$(R$,I,1) THEN T$=LEFT$(T$,I-1)+E
$+MID$(T$,I+1):BEEP
 230 CURSOR I,1:PEN 3:PRINTMID$(T$,I,1):PEN 2
 240 NEXT I
 250 C=C+1
 260 IF T$=R$ THEN CURSOR 0,11:PRINT"TROUVE EN"
;C-1;"COUPS.        ":LASER:STANDARD:END
 270 GOTO 130
```

Nicomaque

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexion (calcul mental).
Difficulté du jeu : moyenne.

LE JEU :
Il utilise une propriété un peu particulière des entiers de 1 à 100 qui peuvent être identifiés sans ambiguïté par leurs restes dans les divisions par 3, 5 et 7.
Le programme, proposé ici, est un programme codeur qui affiche simplement les restes d'un entier donné N dans les divisions indiquées.
Le jeu consiste à essayer de découvrir l'entier (mais, dans cette version, vous n'avez droit qu'à un seul essai).

LE PROGRAMME :
La ligne 50 permet de confectionner l'entier.
La boucle I (80-130) calcule les trois restes successifs, et la boucle emboîtée J (100-120) permet de les afficher sous une forme assez visuelle, aux bons emplacements.
Les lignes 60 et 70 sont là pour indiquer de quelle division il s'agit.
Le nombre proposé par le joueur est entré à la ligne 140 ; il est affiché et testé aux lignes 150 et 160 (affichage de fin de partie : BRAVO ou PERDU).
Les lignes 170 et 180 s'occupent de gérer la prochaine partie, selon la réponse du joueur.

POUR JOUER :
On lance le programme en tapant RUN.
Les trois restes seront alors affichés sur la même ligne, symbolisés par des étoiles (l'absence d'étoiles signifiant : reste 0).
A la question VOTRE ESSAI, on répondra en donnant un entier compris entre 1 et 100, suivi de ENTREE.
On obtiendra, en réponse, un affichage PERDU ou BRAVO, puis la question AUTRE PARTIE (O/N).
Pour faire un nouvel essai, on répondra O, suivi de ENTREE.

EXTENSIONS POSSIBLES :
On peut envisager de réaliser le programme décodeur (on donne les restes et HECTOR devra vous donner N) et chercher si la propriété peut être étendue (en effectuant davantage de divisions, par exemple).

```
10 'NICOMAQUE
20 SEEDTIME(0)
30 PAPER0:PEN2
40 CLS:SPECIAL:SCALE 1,2
50 X=INT(RND(1,101))
60 PRINT"RESTE DANS LA DIVISION PAR"
70 PRINT:PEN1:PRINT"3        5        7":PRINT
:PRINT"..      ....      ......"
80 FOR I=1 TO 3
90 A=X-(2*I+1)*INT(X/(2*I+1))
100 IF A=0 THEN GOTO 130
110 FOR J=1 TO A:CURSOR I*9-10+J,4:PRINT"*"
120 NEXT J
130 NEXT I
140 PEN2:CURSOR 0,7:INPUT"VOTRE ESSAI :";N
150 PEN3:IF N=X THEN PRINT:PRINT"BRAVO ..."
160 IF N<>X THEN PRINT:PRINT"PERDU ... C'ETAIT
";X
170 PRINT:PEN 2:INPUT"AUTRE PARTIE (O/N) :";R$

180 IF LEFT$(R$,1)="O" THEN GOTO 40
190 STANDARD:PEN2
```

Course de chevaux

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : hasard (plusieurs joueurs).
Difficulté du jeu : aucune.

LE JEU :
On entrera un nombre de chevaux compris entre 1 et 4, et la course débutera immédiatement.
Il pourra y avoir des ex aequo, car l'on tiendra compte du fait que HECTOR s'occupe de faire avancer les chevaux les uns après les autres, en commençant par le numéro 1 ; le classement sera affiché au fur et à mesure des arrivées (et en tenant compte des ex aequo).

LE PROGRAMME :
Une première boucle non formelle (8Ø-17Ø) affiche les chevaux dans leur position de départ.
Une deuxième boucle I (1ØØ-18Ø) les fera avancer ; la position de chaque cheval sera repérée par un chiffre ou une lettre dans une variable alphanumérique A$.
La ligne 12Ø fera avancer les chevaux.
La ligne 14Ø gère les arrivées.
La ligne 15Ø réalisera les impressions.
Le sous-programme 24Ø-28Ø fait avancer le compteur des arrivées, et imprime les résultats.

POUR JOUER :
On lance le programme par RUN.
A la question NOMBRE DE CHEVAUX : on répond par un entier de 1 à 4, suivi de ENTREE.
La course se déroulera ensuite automatiquement.

EXTENSIONS POSSIBLES :
On pourra dessiner les chevaux avec un graphisme plus soigné, et les différencier par leur couleur.
On pourra aussi imaginer tout un système de paris, qui interviendra avant le départ de chaque course.

```
10 'COURSE DE CHEVAUX
20 SEEDTIME(0)
30 PAPER0:PEN2
40 CLS:SPECIAL:SCALE 1,2
50 INPUT"NB. DE CHEVAUX :";N:IF N>5 THEN GOTO
40
60 A$="":P=0:A=0:T=1:K=0
70 J=T
80 A$=A$+"0":I=J
90 GOTO 150
100 FOR I=1 TO N
110 IF MID$(A$,I,1)="D" THEN GOTO 180
120 B$=CHR$(ASC(MID$(A$,I,1))+INT(RND(1,6)))
130 A$=LEFT$(A$,I-1)+B$+MID$(A$,I+1)
140 IF B$>="D" THEN GOSUB 240
150 BEEP:CURSOR ASC(MID$(A$,I,1))-48,2*I:PRINT
"     ";I
160 PAUSE.3
170 IF K=0 THEN J=J+1:IF J<=N THEN GOTO 80:ELS
E K=1:GOTO 100
180 NEXT I
190 P=P+T
200 T=0
210 IF A<N THEN GOTO 100
220 STANDARD:PEN2
230 END
240 A$=LEFT$(A$,I-1)+"D"+MID$(A$,I+1)
250 CURSOR 0,2*I:PRINTP
260 T=T+1:LASER
270 A=A+1
280 RETURN
```

A travers

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Sur l'écran apparaît un carré de 9 cases, avec sur le bord vertical des lettres et sur le bord horizontal des chiffres.
L'un des joueurs manipule les chiffres, l'autre les lettres. 3 déplacements sont possibles pour les chiffres, et 3 autres pour les lettres. Les lettres peuvent se déplacer vers le haut, à droite et à gauche et les chiffres vers la droite, en haut ou en bas.
Le gagnant est le premier qui arrive à faire sortir tous ses symboles (les lettres sortent en haut, et les chiffres à droite).
Il n'y a pas de prise, et les sauts ne sont pas autorisés ; on ne peut pas non plus se placer sur une case déjà occupée.

LE PROGRAMME :
Les positions des chiffres et des lettres sont enregistrées dans une table N (4, 4) ; les lignes 10 à 70 définissent les valeurs de départ.
Les lignes 90 à 140 dessinent le terrain de jeu et la ligne 160 affiche le joueur qui a le trait.
La ligne 190 contrôle le coup joué, de même que les boucles I et J (200 à 240), et le changement de joueur est assuré par la ligne 330.

POUR JOUER :
Le terrain de jeu sera affiché après la frappe de RUN et l'ordinateur attend le coup du joueur désigné.
On entrera d'abord le chiffre ou la lettre représentant le pion à déplacer, suivi d'une indication de direction sous la forme N (pour Nord), S (pour Sud), E (pour Est) ou O (pour Ouest).
On pourra ainsi faire 2E suivi de ENTREE.
Si le coup est invalide, il ne sera pas exécuté, et le code du joueur restera à l'affichage ; s'il est autorisé, le déplacement sera effectué, et l'on obtient l'affichage correspondant au deuxième joueur.
Le premier qui aura réussi à faire disparaître ses 3 pièces aura gagné.

EXTENSIONS POSSIBLES :
On peut envisager d'étendre le jeu avec davantage de chiffres ou de lettres et d'améliorer les contrôles. On peut aussi définir une stratégie pour HECTOR.

```
 10 'A TRAVERS
 20 CLS:PEN 2:SPECIAL:SCALE 1,2
 30 DIM N(4,4):K=1
 40 FOR I=1 TO 4
 50 N(I,1)=ASC("0")+I:N(4,I)=ASC("?")+I
 60 NEXT I
 70 N(4,1)=0
 80 CLS:SCALE 1,2
 90 FOR U=1 TO 4
 100 FOR V=1 TO 4
 110 CURSOR 4*U,3*V-3
 120 IF N(V,U)=0 THEN PEN2: PRINT".":ELSE PEN 1
-2*(N(V,U)<58):PRINTCHR$(N(V,U))
 130 NEXT V
 140 NEXT U
 150 SCALE 1,1:CURSOR 0,23:PEN 2
 160 PRINT"JOUEUR AYANT LES ";:IF K=1 THEN PRIN
T"CHIFFRES":ELSEPRINT"LETTRES"
 170 PRINT:INPUT"VOTRE COUP (N,S,E,O) :";R$
 190 IF (K=1 AND RIGHT$(R$,1)="O") OR (K=0 AND
RIGHT$(R$,1)="S") THEN BEEP:GOTO 150
 200 FOR I=1 TO 4
 210 FOR J=1 TO 4
 220 IF N(I,J)=ASC(LEFT$(R$,1)) THEN A=I:B=J:I=
4:J=4:NEXT J,I:GOTO 260
 230 NEXT J
 240 NEXT I
 250 GOTO 150
 260 I=A:J=B
 270 I=I-(RIGHT$(R$,1)="S")+(RIGHT$(R$,1)="N")
 280 J=J-(RIGHT$(R$,1)="E")+(RIGHT$(R$,1)="O")
 290 IF (I=0 AND K=0) OR (J>4 AND K=1) THEN GOT
O 320
 300 IF N(I,J)<>0 THEN BEEP:GOTO 150
 310 N(I,J)=ASC(LEFT$(R$,1))
 320 N(A,B)=0
 330 K=ABS(K-1)
 340 GOTO 80
```

Mastermind

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : logique (un seul joueur).
Difficulté du jeu : difficile.

LE JEU :
Devenu un grand classique des jeux sur micro-ordinateur, ce jeu développe le raisonnement déductif ; il faut réussir à trouver le plus vite possible un nombre de n chiffres (on pourra prendre n compris entre 1 et 4), en disposant à chaque essai de renseignements.
En effet, HECTOR répondra en indiquant d'une part le nombre de chiffres qui sont, dans le nombre proposé, exactement à la même place que dans le nombre à découvrir, et, d'autre part, le nombre de chiffres qui figurent dans le nombre à découvrir sans être exactement au bon emplacement.

LE PROGRAMME :
Le choix du nombre de chiffres du nombre à découvrir est effectué aux lignes 4Ø et 5Ø.
Le nombre est fabriqué à la ligne 7Ø (il pourra y avoir des chiffres répétés).
L'affichage est préparé par la ligne 9Ø.
La ligne 13Ø s'occupe de la fin de partie (lignes 27Ø-28Ø).
Une première boucle I (14Ø-23Ø) teste les chiffres qui sont à la bonne place.
Une deuxième boucle J (16Ø-28Ø) teste les chiffres qui sont présents et mal placés.
Les compteurs sont aux lignes 2ØØ et 22Ø.
La ligne 24Ø affiche le résultat, et la ligne 25Ø comptabilise les essais.

POUR JOUER :
On tape RUN et l'on choisit la longueur du nombre désiré (de 1 à 4), suivie de ENTREE.
Après l'affichage NO. NOMBRE BON MAL, on introduit son propre essai, suivi de ENTREE.
On obtient immédiatement la réponse (en cas de double chiffre, les résultats peuvent être un peu délicats à interpréter).
On introduit ensuite le nouvel essai, et ainsi de suite.
Lorsque l'on a trouvé, on obtiendra l'affichage TROUVE EN ... COUPS.
On peut sortir du programme à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
On peut chercher à éviter la présence de chiffres doubles dans le nombre à trouver, on peut aussi prévoir un changement de page automatique, avec rappel des résultats précédents ; on pourra aussi réaliser des jeux analogues avec des mots, ou d'autres symboles (couleur, taille...).

```
 10 'MASTERMIND
 20 CLS:SPECIAL:SCALE1,2
 30 SEEDTIME(0)
 40 INPUT"NB. DE CHIFFRES :";C
 50 IF C>5 THEN GOTO 40
 60 E=1:F=1
 70 X$=RIGHT$(STR$(100000+INT(RND(0,10^C+1))),C
)
 80 CLS
 90 CURSOR 0,0:PRINT"ESSAI N.  NB.ESSAYE BON MA
L"
 100 CURSOR 0,1:PRINT"__________________________
"
 110 CURSOR 0,E+2:PRINTTAB(4);F;TAB(10);:INPUT"
";E$
 120 X=0:Y=X
 130 IF E$=X$ THEN GOTO 270
 140 FOR I=1 TO C
 150 IF MID$(X$,I,1)=MID$(E$,I,1) THEN GOTO 220

 160 FOR J=1 TO C
 170 IF J<>I AND MID$(X$,I,1)=MID$(E$,J,1) THEN
 J=C:NEXT J:GOTO 200
 180 NEXT J
 190 GOTO 230
 200 Y=Y+1
 210 GOTO 230
 220 X=X+1
 230 NEXT I
 240 CURSOR 20,E+2:PRINTX;" ";Y:BEEP
 250 E=E+1:F=F+1:IF E>10 THEN E=1
 260 GOTO110
 270 LASER:PRINT:PRINTX$;" TROUVE EN";F;"COUPS.
"
 280 STANDARD
```

Idem

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de calcul mental (2 joueurs).
Difficulté du jeu : selon le niveau choisi.

LE JEU :
HECTOR choisit un nombre de n chiffres (selon le niveau de difficulté indiqué au début). Ce nombre (le même au début pour les 2 joueurs) sera régulièrement décrémenté à chaque tour, d'une quantité également choisie au hasard par HECTOR en début de jeu ; vous devez, à chaque coup, choisir une colonne (les dizaines, les centaines, etc.) et ajouter dans cette colonne un certain entier ; le but est d'arriver à obtenir à un moment l'affichage d'un nombre ayant tous ses chiffres identiques.
Chaque résultat sera affiché un court instant, après quoi la main passera au joueur suivant, qui repartira de son précédent résultat.

LE PROGRAMME :
Une variable numérique N est dimensionnée à 2 en ligne 4Ø, pour contenir les résultats successifs des deux joueurs.
La ligne 5Ø permettra d'entrer le niveau de difficulté choisi.
Le nombre initial est confectionné par la ligne 7Ø. La quantité qui interviendra dans les décrémentations est obtenue en ligne 9Ø.
La ligne 12Ø teste si le résultat n'est pas trop petit.
La ligne 13Ø affiche le numéro du joueur.
Le nombre est affiché en ligne 14Ø et le jeu du joueur est entré en ligne 16Ø (nombre de deux chiffres).
La ligne 17Ø incrémente la colonne choisie, et la ligne 18Ø décrémente le nombre de la quantité choisie au départ.
Le résultat sera affiché un court instant (ligne 21Ø). La boucle I (26Ø-28Ø) permet de vérifier si tous les chiffres sont identiques.
L'impression et la fin de partie sont effectuées en ligne 29Ø. Les lignes 3ØØ et 31Ø s'occupent du changement de joueur.

POUR JOUER :
Lancer l'exécution du programme par RUN.
Le nombre initial sera affiché pour le joueur n° 1, qui joue en tapant xy, suivi de ENTREE ; x est le chiffre à ajouter, et y est le numéro de la colonne où il faut l'ajouter.
Le résultat (provisoire s'il est trop petit) sera alors affiché un court instant, et le trait passe au deuxième joueur, et ainsi de suite. On peut arrêter à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
On pourra étendre le jeu à plus de 2 joueurs, améliorer le choix du nombre intervenant dans la décrémentation, améliorer les divers contrôles, laisser les différents résultats affichés, etc.

```
 10 'IDEM
 20 CLS:SPECIAL:SCALE1,2
 30 SEEDTIME(0)
 40 CLS:H=1:DIM N(2)
 50 INPUT"NIVEAU (2-7) :";N
 60 U=10:P=0
 70 N(1)=INT((U^N-U^(N-1))*RND(0,1)+U^(N-1))
 80 N(2)=N(1)
 90 A=INT(2*U^(N-1)*RND(0,1)+U^(N-1))
 100 IF H=1 THEN P=P+1
 110 CLS
 120 IF N(H)<U^(N-1) THEN N(H)=U^N-ABS(N(H))
 130 PRINT"JOUEUR";H;" ESSAI";P
 140 CURSOR 0,2:PRINT"NOMBRE :";:CURSOR 15-N,2:
PEN 1:PRINTN(H):PEN2
 150 CURSOR 9,3:PRINT"7654321"
 160 CURSOR 0,6:INPUT"VOTRE JEU (NC) :";R$
 170 N(H)=N(H)+VAL(LEFT$(R$,1))*U^(VAL(RIGHT$(R
$,1))-1)
 180 N(H)=N(H)-A
 190 IF N(H)>U^N THEN N(H)=N(H)-U^N
 200 N(H)=INT(N(H))
 210 CURSOR 0,9:PRINT"RESULTAT ";:SCALE 2,2:CUR
SOR 4,9:PEN1:PRINTN(H):SCALE 1,2:PEN2
 220 SHOTO
 230 PAUSE 1.2
 240 R$=STR$(N(H))
 250 R$=MID$(R$,2)
 260 FOR I=1 TO LEN(R$)-1
 270 IF MID$(R$,I,1)<>MID$(R$,I+1,1) THEN I=LEN
(R$):NEXT I:GOTO 300
 280 NEXT I
 290 CURSOR 0,11:PRINT"BRAVO, JOUEUR";H:LASER:S
TANDARD:END
 300 H=H+1
 310 IF H>=3 THEN H=1
 320 GOTO 100
```

Horloge 1

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : dessin.
Difficulté du jeu : aucune.

LE JEU :
Transformer votre écran de télévision en horloge numérique à affichage grand format, grâce à ce programme qui vous permettra d'animer une vitrine, de faire une horloge fantaisie, et qui pourra se prêter à bien d'autres usages.

LE PROGRAMME :
La ligne 3Ø vous permettra d'entrer l'heure sous forme d'un entier de 4 chiffres (par exemple : 1Ø24 signifera 1Ø heures 24 minutes).
Les lignes 8Ø-17Ø réalisent le dessin, en décalant l'impression au fur et à mesure (à l'aide de la variable K, incrémentée en ligne 16Ø).
Les lignes 18Ø à 22Ø assurent la progression de l'heure.
Les lignes 23Ø à 29Ø effectuent l'affichage de deux points clignotants pour les secondes.

POUR JOUER :
Dès le lancement par RUN, le programme se place en attente de l'heure ; on tape et on attend le top pour appuyer sur ENTREE.
L'heure s'affiche alors assez rapidement et l'horloge est en route. Pour l'arrêter, il faudra appuyer sur les touches CNT et C.
L'heure sera entrée sous forme d'un entier ayant impérativement 4 chiffres, pour 5 heures 6 minutes, on veillera à taper Ø5Ø6 suivi de ENTREE.

EXTENSIONS POSSIBLES :
Il sera intéressant de modifier le programme pour en faire un réveil, avec alarme ; on peut aussi améliorer le graphisme, introduire la couleur.

```
10 'HORLOGE 1
20 CLS:SPECIAL:SCALE1,2
30 INPUT"HEURE (FORME HHMM) :";A$
40 CLS1:PAPER 3
50 K=1
60 A=VAL(MID$(A$,K,1))
70 IF A<>1 AND A<>4 THEN CURSOR 6*K-5,1:PRINT"
 "
80 FOR J=1 TO 5
90 IF A=0 OR (A>3 AND A<>7) THEN CURSOR 6*K-5,
J:PRINT" "
100 IF A<>5 AND A<>6 THEN CURSOR 6*K-3,J:PRINT
" "
110 IF A=0 OR A=2 OR A=6 OR A=8 THEN CURSOR 6*
K-5,J+4:PRINT" "
120 IF A<>2 THEN CURSOR 6*K-3,J+4:PRINT" "
130 NEXT J
140 IF A>1 THEN CURSOR 6*K-4,5:PRINT"  "
150 IF A<>1 AND A<>4 AND A<>7 THEN CURSOR 6*K-
5,9:PRINT"   "
160 K=K+1
170 IF K<5 THEN GOTO 60
180 A=1+VAL(A$)
190 IF A=2360 THEN A=0
200 A$=STR$(A)
210 A$=RIGHT$("000"+MID$(A$,2),4)
220 IF RIGHT$(A$,2)="60"THENA=A+40:GOTO200
230 H=1
240 FOR J=1 TO 120
250 FOR I=1 TO 47
260 CURSOR 11,9:PEN0:IF H=1 THEN PRINT":":ELSE
PRINT" "
270 NEXT I
280 H=ABS(H-1):IF H=1 THEN BEEP
290 NEXT J
300 LASER:GOTO 40
```

Reines sur l'échiquier

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :
Le programme utilise un algorithme très performant pour résoudre un problème qui a donné du fil à retordre à des chercheurs, même aussi célèbres que le mathématicien GAUSS, qui n'avait pas pu découvrir toutes les solutions. Il s'agit de trouver toutes les manières de disposer des reines sur un échiquier, de telle sorte qu'aucune ne puisse en prendre une autre. Le programme proposé est très souple, puisque vous disposez même de la possibilité de choisir la taille de votre échiquier, et les résultats seront affichés à la fois sous forme visuelle et sous forme numérique.

LE PROGRAMME :
La ligne 3Ø vous permet de choisir la taille de l'échiquier.
Une variable D sera dimensionnée en conséquence, en ligne 4Ø.
Une boucle d'exploration principale I (5Ø-11Ø) contient une sous-boucle J (8Ø-1ØØ) qui réalise le contrôle de non-alignement (ligne 9Ø).
Si, au cours de l'exploration, on rencontre un alignement, on revient en arrière, en utilisant les lignes 23Ø à 27Ø et on recommence.
La boucle F (13Ø-21Ø) assure l'impression de l'échiquier, et la boucle K (16Ø-18Ø), que l'on pourra éventuellement supprimer, assure l'impression du résultat sous forme d'un code numérique correspondant à la position des pièces dans chaque colonne.
La ligne 22Ø permet un arrêt après l'affichage de chacune des solutions.

POUR JOUER :
On lance le programme avec RUN, puis on tape la dimension de l'échiquier désirée, et on fait ENTREE.
Il suffit d'attendre, avec patience (plus l'échiquier sera grand, plus l'attente sera longue). La première solution sera affichée puis le programme s'arrête, et il faut appuyer sur la touche ENTREE pour le faire repartir. Pour terminer l'exploration sans avoir eu toutes les solutions, il faudra taper CNT et C.

EXTENSIONS POSSIBLES :
On peut envisager de trouver des algorithmes pour tous les problèmes analogues (position de tours, déplacement d'un cavalier, etc.).

```
10 'REINES SUR L'ECHIQUIER
20 CLS:SPECIAL:SCALE1,1
30 INPUT"TAILLE DE L'ECHIQUIER :";S
40 DIM D(S)
50 I=1
60 D(I)=1
70 IF I=1 THEN GOTO 110
80 FOR J=1 TO I-1
90 IF D(I)=D(J) OR ABS(D(I)-D(J))=I-J THEN J=I
:NEXT J:GOTO 240
100 NEXT J
110 I=I+1:IF I<=S THEN GOTO 60
120 CLS
130 FOR F=1 TO S
140 CURSOR 2*F,0:PEN 2:PRINTF
150 CURSOR 0,2*F:PRINTF
160 FOR K=1 TO S
170 CURSOR 2*K+1,2*F:PRINT"."
180 NEXT K
190 CURSOR 2*D(F)+1,2*F:PEN1:PRINT"*"
200 CURSOR F,20:PEN 3:PRINTMID$(STR$(D(F)),2)
210 NEXT F
220 LASER:PEN 2:PRINT:PRINT:INPUT"TAPER 'ENTRE
E' POUR SUITE";R$
230 I=I-1
240 D(I)=D(I)+1
250 IF D(I)<=S THEN GOTO 70
260 I=I-1
270 IF I<>0 THEN GOTO 240
280 STANDARD
```

Répétitions

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de mémoire (1 ou plusieurs joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Des nombres vont apparaître un par un.
Le joueur devra à chaque coup les répéter tous, depuis le début (ce qui signifie qu'il en aura un de plus à taper à chaque fois).
Il devra essayer de tenir le plus longtemps possible. Le meilleur score est mémorisé après chaque partie, ce qui permettra éventuellement de jouer à plusieurs.

LE PROGRAMME :
La ligne 6Ø appelle un sous-programme (lignes 1ØØØ à 1Ø6Ø) qui donne la règle du jeu.
La ligne 8Ø fabrique les chiffres, qui sont ensuite stockés dans une variable alphanumérique A$.
La ligne 9Ø permet l'impression du dernier chiffre au hasard sur la quatrième ligne de l'écran, et la durée de l'affichage est déterminée par la ligne 1ØØ.
La ligne 12Ø attend la réponse du joueur, qui est comparée à la réponse correcte en ligne 13Ø.
La ligne 16Ø marque les points.
La ligne 17Ø contrôle le meilleur score.
Les affichages de fin de partie sont effectués de la ligne 2ØØ à la ligne 24Ø.

POUR JOUER :
Le lancement du programme par RUN provoque l'affichage de la règle du jeu, puis d'un chiffre qui disparaît. L'appareil pose alors la question VOTRE SEQUENCE.
On lui répond en redonnant le chiffre, suivi de ENTREE. Si c'est exact, on a un court instant l'affichage de EXACT. ATTENTION, puis un nouveau chiffre est affiché, qui s'efface, et de nouveau la question VOTRE SEQUENCE. On retape alors les deux chiffres déjà obtenus, suivis de ENTREE.
On continue ainsi (3 chiffres, puis 4, 5, etc.) jusqu'à ce que l'on fasse une erreur, ce qui provoque l'affichage du score (et du meilleur score obtenu jusqu'à présent), suivi de la question AUTRE PARTIE (O/N) ; on tape O pour continuer.
On peut aussi quitter le programme à tout moment, en tapant CNT et C avant l'affichage d'un chiffre.

EXTENSIONS POSSIBLES :
On pourra réaliser des jeux analogues et plus ou moins perfectionnés en remplaçant les chiffres par des lettres ou des couleurs ou des sons...

```
 10 'REPETITIONS
 20 CLS:PEN2
 30 SEEDTIME(0)
 40 M=0
 50 A$=""
 60 GOSUB 1000
 70 H=0
 80 X=INT(RND(0,10)):A$=A$+MID$(STR$(X),2)
 90 CLS:PEN1:SCALE 2,2:CURSOR RND(0,13),3:PRINT
X:PEN2
 100 PAUSE .2+RND(0,.3)
 110 CLS:SCALE 1,2
 120 CURSOR 0,9:INPUT"VOTRE SEQUENCE :";R$
 130 IF R$<>A$ THEN GOTO 190
 140 LASER:PRINT:PRINT"EXACT...ATTENTION"
 150 PAUSE1
 160 H=H+1
 170 IF M<H THEN M=H
 180 GOTO 80
 190 CLS:SIREN
 200 PRINT"VOTRE SCORE EST DE";H
 210 PRINT:PRINT"MEILLEUR SCORE :";M
 220 PRINT:INPUT"AUTRE PARTIE (O/N) :";R$
 230 HUSH:IF R$="N" THEN SHOT1:STANDARD:END
 240 GOTO 50
 1000 STANDARD:CLS
 1010 PRINT"VOUS DEVEZ REPETER TOUTE LA SEQUENC
E"
 1020 PRINT:PRINT"DES NOMBRES APPARUS DEPUIS LE
 DEBUT"
 1030 PRINT:PRINT:PRINT"IL Y EN A UN DE PLUS A
CHAQUE FOIS !"
 1040 PRINT:PRINT:INPUT"APPUYER SUR 'ENTREE' PO
UR COMMENCER";R$
 1050 SPECIAL:SCALE 1,2
 1060 RETURN
```

Mots

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de lettres (pour 1 ou plusieurs joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Il est difficile d'imaginer plus simple dans son principe que ce jeu, qui se contente d'afficher dans le désordre les lettres prises dans une trentaine de mots. Mais essayez de jouer, et vous verrez que ce n'est pas si facile, la difficulté variant évidemment selon la liste de mots que vous aurez introduits en DATA au préalable.

LE PROGRAMME :
La ligne 4Ø permet de dimensionner la variable A$ (I) qui contiendra les mots.
La boucle I (5Ø) saisit la liste des mots (inscrits en DATA aux lignes 24Ø à 28Ø).
La ligne 7Ø choisit un mot au hasard ; la boucle I (1ØØ-15Ø) contrôle si une lettre n'a pas déjà été tirée, et reprend les lettres dans le désordre.
La ligne 16Ø affiche les lettres, et la ligne 17Ø attend votre proposition, qui est testée en ligne 18Ø.
Le nombre d'essais est incrémenté en ligne 19Ø.
La fin de partie est contrôlée par la ligne 2ØØ, et le score est affiché en ligne 22Ø.

POUR JOUER :
On lance le programme par RUN. Une série de lettres est affichée dans le désordre, et il faut essayer de retrouver le mot correspondant.
On tape le mot supposé, puis ENTREE, et une nouvelle série de lettres sera affichée, si l'on répond O à la question AUTRE ESSAI (O/N).
Le score et le nombre d'essais sont affichés en fin de partie.

EXTENSIONS POSSIBLES :
On peut imaginer de minuter le temps d'apparition du mot, ou de faire défiler les lettres l'une après l'autre sur l'écran.
On peut aussi corser la difficulté en introduisant des JOKERS, jouer à plusieurs, tenir le compte des points, etc.

```
10 'MOTS
15 CLEAR 1000
20 CLS:PEN2
30 SEEDTIME(0)
40 DIM A$(30):S=0:E=0
50 FOR I=1 TO 30:READ A$(I):NEXT I
60 CLS
70 Z=INT(RND(1,31))
80 IF A$(Z)="" THEN GOTO 70
90 R$=A$(Z):A$="":A$(Z)="":V$=R$
100 FOR I=1 TO LEN(R$)
110 Z=INT(RND(1,LEN(R$)+1))
120 IF MID$(R$,Z,1)="-" THEN GOTO 110
130 A$=A$+MID$(R$,Z,1)
140 R$=LEFT$(R$,Z-1)+"-"+MID$(R$,Z+1)
150 NEXT I
160 PRINT"LES LETTRES SONT :";A$:BEEP
170 PRINT:PRINT:PRINT:PRINT:PRINT:INPUT"VOTRE
SOLUTION :";S$
180 IF S$=V$ THEN S=S+1
190 E=E+1
200 PRINT:PRINT:PRINT:PRINT:PRINT:INPUT"AUTRE
ESSAI (O/N) :";R$
210 IF LEFT$(R$,1)="O" THEN GOTO 60
220 PRINT:PRINT"SCORE :";S;"SUR";E;"ESSAIS"
230 STANDARD:END
240 DATAJAMBON,ENERGUMENE,AFFILIE,ERUDIT,INSIG
NIFIANT,REALISTE
250 DATAIMPRESSIONNANT,ALLECHANT,EPIQUE,INTREP
IDE,COURD,ORDINATEUR
260 DATATREVE,BANDEROLLE,SENAT,ACTUEL,CONSACRE
,DIEDRE
270 DATALIVRE,TERMINE,SIGNATURE,MALODORANT,AIM
ANT,ARBITRAIRE
280 DATACHALEUR,OPPORTUN,PELURE,ALTRUISTE,RAMO
NEUR,NOVICE
```

Permutations

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : moyenne.

LE JEU :
HECTOR vous demande d'abord le nombre de lettres que vous désirez permuter (pas plus de 12). A partir de là, il va vous afficher le nombre correspondant de lettres prises dans le début de l'alphabet, d'abord dans le désordre (DEBUT) puis dans l'ordre (FIN).
Vous devez réussir à remettre dans l'ordre les lettres données au début, en utilisant des permutations sur les n premières lettres à partir de la gauche, avec le moins d'essais possible.

LE PROGRAMME :
La ligne 40 permet de choisir le nombre de lettres, et la ligne 50 contrôle que l'on n'en demande pas plus de 12.
La ligne 80 réalise le brouillage, en tirant au hasard la nouvelle position de chaque lettre. Les lettres sont ensuite ajoutées au fur et à mesure de leur tirage, dans une variable B$ (ligne 100), supprimées dans A$ et replacées en C$, dans le bon ordre (lignes 110-120).
La ligne 140 contrôle le nombre de lettres de B$.
Les affichages sont effectués aux lignes 160, 170 et 180.
Le retournement est fait, lettre par lettre, dans la boucle I (210-230) ; le résultat est placé dans B$ en ligne 240, affiché par la ligne 250 et comparé au résultat final en ligne 260.
Le nombre d'essais sera affiché en fin de partie, grâce à la ligne 270.

POUR JOUER :
Après avoir tapé RUN, on obtient l'affichage de la question NOMBRE DE LETTRES (DE 1 A 12) ; on répond en frappant N, suivi de ENTREE.
L'affichage suivant présente l'ordre initial (DEBUT) et l'ordre à obtenir (FIN).
On joue en indiquant le nombre de lettres à permuter à partir de la gauche, par exemple : sur un affichage initial de GEABDFC, si l'on tape 3 et ENTREE, on obtiendra AEGBDFC.
On poursuivra jusqu'à l'obtention de l'ordre désiré.
Le score sera affiché en fin de partie ainsi qu'une question AUTRE PARTIE (O/N) ; répondre O si l'on désire faire une autre partie.

EXTENSIONS POSSIBLES :
On pourra utiliser le même type de jeu pour, par exemple, reconstituer un dessin caché dont les diverses parties seraient repérées par des lettres. On peut aussi réaliser des permutations plus complexes (par exemple le fait de permuter deux lettres permute les deux voisines de la lettre la plus à gauche, etc.).

```
 10 'PERMUTATIONS
 20 CLS:PEN2:COVER 1
 30 SEEDTIME(0)
 40 INPUT"NOMBRE DE LETTRES (1-12) :";N
 50 IF N>12 OR N<1 THEN GOTO 40
 60 A$="ABCDEFGHIJKL":C$="            ":B$=""
 70 CLS
 80 X=INT(RND(1,N+1))
 90 IF MID$(A$,X,1)=" " THEN GOTO 80
 100 B$=MID$(A$,X,1)+B$
 110 C$=LEFT$(C$,X-1)+MID$(A$,X,1)+MID$(C$,X+1)

 120 A$=LEFT$(A$,X-1)+" "+MID$(A$,X+1)
 130 S=1
 140 IF LEN(B$)<>N THEN GOTO 80
 150 C$=LEFT$(C$,LEN(B$))
 160 PRINT"DEBUT ... :";B$
 170 PRINT:PRINT"FIN ..... :";C$
 180 CURSOR9,160:PRINT:PRINT:PRINT:PRINT"PERMUT
ATION NUMERO ";S:PRINT:PRINT:INPUT"VOTRE QUANTI
TE :";L:IF L>N THEN BEEP:GOTO180
 190 S=S+1
 200 A$=""
 210 FOR I=1 TO L
 220 A$=A$+MID$(B$,L-I+1,1)
 230 NEXT I
 240 B$=A$+MID$(B$,L+1)
 250 PRINT:PRINT:PRINT"RESULTAT :";:PEN1:PRINTB
$:PEN2
 260 IF B$<>C$ THEN GOTO 180
 270 LASER:PRINT:PRINT:PRINT"TROUVE EN";S-1;"ES
SAIS..."
 280 PRINT:INPUT"AUTRE PARTIE (O/N) :";R$
 290 IF LEFT$(R$,1)="O" THEN CLS:GOTO 40
```

Bridge

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : variable, selon le problème posé.

LE JEU :
Vous ne trouverez pas, dans ce programme, un partenaire pour vos bridges mais, plus exactement, une simulation pour l'affichage d'une donne ; cet affichage est réalisé jeu par jeu. On peut, à partir de là, se poser toutes sortes de problèmes : annonces, jeu de la carte, etc.

LE PROGRAMME :
Les lignes 5Ø et 6Ø définissent un jeu de 52 cartes.
La variable B$, en ligne 7Ø, permettra de donner en abrégé le nom des couleurs et aussi d'améliorer la désignation de certaines cartes (1Ø, valet, dame, roi et as).
La boucle J (8Ø-22Ø) réalise le tirage de quatre jeux de 13 cartes.
La boucle I (9Ø-19Ø) réalise tirage et impression. La carte est choisie parmi les 52 disponibles, en ligne 1ØØ.
La ligne 11Ø calcule la couleur de la carte tirée.
La ligne 12Ø vérifie que cette carte n'a pas déjà été tirée.
Les lignes 13Ø et 14Ø s'occupent de l'impression complémentaire dans le cas où l'on tire un 1Ø, un valet, une dame, un roi ou un as.
L'impression est réalisée par les lignes 15Ø, 16Ø et 17Ø.
La ligne 18Ø permet d'éliminer les cartes déjà tirées, et la ligne 2ØØ réalise un arrêt entre chaque affichage.

POUR JOUER :
On lance le programme par RUN.
Une première série de 13 cartes est affichée verticalement. Il suffira d'appuyer sur la touche ENTREE pour obtenir la deuxième série de 13 cartes. On procédera de même pour la troisième et la quatrième.

EXTENSIONS POSSIBLES :
On peut imaginer de représenter les 13 cartes sous une forme plus conventionnelle.
On peut aussi ajouter la possibilité de faire des annonces (et de les comparer à celles de HECTOR, par exemple), ou introduire des possibilités de jeu de la carte, y compris pour HECTOR.

```
 10 'BRIDGE
 20 COVERO
 30 SEEDTIME(0)
 40 CLS
 50 R$="234567891VDRA"
 60 R$=R$+R$+R$+R$
 70 B$="TRCACOPIOAAOS"
 80 FOR J=1 TO 4
 90 FOR I=1 TO 13
 100 X=INT(RND(1,LEN(R$)+1))
 110 S=2*INT((X-1)/13)+1
 120 IF MID$(R$,X,1)="0" THEN GOTO 100
 130 T=X-13*INT(X/13)
 140 T=T-13*(T=0)
 150 PRINT TAB(9*(J-1));MID$(R$,X,1);
 160 IF T>=9 THEN PRINTMID$(B$,T,1);:ELSE PRINT
" ";
 170 PRINT":";MID$(B$,S,2)
 180 R$=LEFT$(R$,X-1)+"0"+MID$(R$,X+1)
 190 BEEP:NEXT I
 200 PRINT:PRINT:INPUT"FAIRE 'ENTREE' POUR CONT
INUER";S$
 210 IF J<>4 THEN HOME
 220 NEXT J
```

Hanoï

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :
Il est présenté, ici, dans une version pour un joueur. On dispose d'un certain nombre de disques (6 au plus) et l'on doit les transférer de la tour numéro 1 à la tour numéro 3, en respectant une seule règle : à aucun moment, un disque ne peut être recouvert par un disque de diamètre supérieur.
Le nombre de coups n'est pas comptabilisé par ce programme mais, par contre, les coups illicites seront refusés.

LE PROGRAMME :
La variable D$, en ligne 4Ø, servira à dessiner les disques.
Le nombre de disques est entré en ligne 5Ø et refusé s'il y a plus de 8 disques.
La boucle I (7Ø-9Ø) permet de fabriquer une variable A (1) qui contient l'état de la première pile sous forme numérique (en ligne 3Ø, trois variables sont prévues pour conserver l'état des trois piles).
Les socles sont dessinés par la ligne 11Ø et les trois piles par les boucles emboîtées I (13Ø-18Ø) et J (14Ø-17Ø).
Le contrôle du coup est effectué par la ligne 26Ø et l'état des piles est modifié par les lignes 27Ø et 28Ø.

POUR JOUER :
Après un RUN qui lance le programme, on entre le nombre de disques demandé, suivi de ENTREE. On obtient le dessin de l'état initial et, à la question TOUR DE DEPART, on répond en donnant le numéro de la tour de départ. Ce numéro (1, 2 ou 3) sera refusé si la tour est vide. Puis on tape ENTREE et, à la question TOUR D'ARRIVEE, on répond par le numéro de la tour d'arrivée.
Si le coup n'est pas possible, on sera obligé de recommencer à TOUR DE DEPART. Si le coup est possible, le déplacement est réalisé et l'on recommence.
Pour arrêter, taper Ø en TOUR DE DEPART.

EXTENSIONS POSSIBLES :
On pourra dessiner des disques de différentes couleurs, introduire un compteur de coups et, surtout, définir un jeu pour l'ordinateur.
On peut aussi essayer de compliquer le jeu avec deux tiges de plus et deux joueurs, chacun ayant des disques de couleur différente, par exemple.

```
 10 'HANOI
 20 CLS:SPECIAL:PEN 2:SCALE 1,2
 30 DIM A$(9):A$(1)="7":A$(2)="7":A$(3)="7"
 40 D$="":FOR I=1 TO 7:D$=D$+"#":NEXT I
 50 INPUT"NB. DE TOURS (<6) :";N
 60 IF N>=6 OR N<1 THEN BEEP:GOTO 50
 70 FOR I=0 TO N-1
 80 A$(1)=A$(1)+MID$(STR$(N-I),2)
 90 NEXT I
 100 CLS
 110 CURSOR0,8:PRINTD$:CURSOR9,8:PRINTD$:CURSOR
 18,8:PRINTD$
 120 PEN 1:CURSOR 3,8:PRINT"1":CURSOR 12,8:PRIN
T"2":CURSOR 21,8:PRINT"3"
 130 FOR I=0 TO N-1
 140 FOR J=1 TO 3
 150 B$=A$(J)
 160 IF LEN(B$)>I+1 THEN CURSOR 9*J-8,7-I:PEN 3
:PRINTLEFT$(D$,VAL(MID$(B$,I+2,1)))
 170 NEXT J
 180 NEXT I
 190 PEN 2
 200 BEEP:SCALE 1,1:CURSOR 0,21:INPUT"TOUR DE D
EPART :";D:IF D=0 THEN GOTO 300
 210 IF D>3 OR D<1 THEN GOTO 200
 220 IF A$(D)="7" THEN GOTO 200
 230 PRINT:INPUT"TOUR D'ARRIVEE :";A
 240 B$=A$(D)
 250 C$=A$(A)
 260 IF MID$(B$,LEN(B$),1)>MID$(C$,LEN(C$),1) T
HEN GOTO 200
 270 A$(A)=C$+MID$(B$,LEN(B$),1)
 280 A$(D)=LEFT$(B$,LEN(B$)-1)
 290 SCALE 1,2:GOTO 100
 300 STANDARD
```

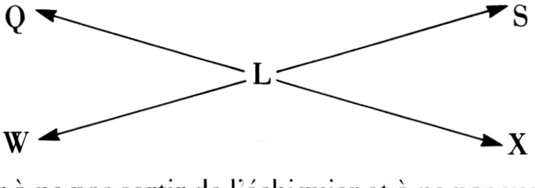
Le loup et les agneaux

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :
C'est un jeu assez connu qui peut être pratiqué indifféremment sur un damier 8 sur 8, ou 1Ø sur 1Ø. Il est présenté, ici, dans une version pour un damier de 8 cases ; l'un des joueurs choisit le loup qui se déplace suivant les diagonales, en restant sur les cases noires, d'une case à chaque coup, et qui peut avancer ou reculer. Les agneaux se déplacent également sur les cases noires, en diagonale, mais n'ont pas le droit de reculer. Les cases déjà occupées sont interdites (pas de prises), et il n'y a pas de saut.
Le loup gagne s'il réussit à arriver sur la ligne du bas avant que tous les agneaux n'occupent celle du haut ; les agneaux gagnent dans ce dernier cas, ou s'ils arrivent à bloquer le loup.

LE PROGRAMME :
Une variable A$, en ligne 3Ø contient les positions de départ.
La boucle I (8Ø-13Ø) dessine le damier, et les lignes 15Ø à 19Ø placent le loup et les agneaux.
La ligne 2ØØ gère le changement de joueur.
La ligne 22Ø affiche JEU DU LOUP si c'est le joueur qui manipule le loup qui a le trait, et la ligne 23Ø JEU DES AGNEAUX dans le cas contraire.
Les lignes 25Ø à 27Ø analysent la réponse, en fonction du joueur concerné.

POUR JOUER :
RUN déclenche l'arrêt sur l'attente d'une entrée : Ø et ENTREE si l'on veut que les agneaux commencent, et 1 pour le loup. Après quoi, le damier s'affiche et la phrase JEU DU LOUP (Q, S, W, X) pour le loup et JEU DES AGNEAUX (ND et NG) pour les agneaux.
Les agneaux répondent par un numéro, suivi de D ou G (pour droite ou gauche).
Ainsi 2D provoque le déplacement de l'agneau numéro 2 qui monte en diagonale d'une case sur la droite. Le déplacement effectué est affiché, puis on obtient une question identique pour le loup qui répondra en tapant Q, S, W ou X, suivant la direction choisie :



On devra veiller à ne pas sortir de l'échiquier et à ne pas venir sur l'adversaire, aucun contrôle n'ayant été prévu.

EXTENSIONS POSSIBLES :

On pourra prévoir un repérage des cases, un graphisme plus soigné, des contrôles de non dépassement des limites du terrain, de non venue sur une case déjà occupée...

On pourra aussi envisager de jouer sur un damier de 1Ø cases sur 1Ø cases, ce qui augmente légèrement la difficulté du jeu pour les agneaux. Mais l'extension la plus intéressante à réaliser sera certainement la réalisation d'un programme permettant à HECTOR de se comporter en adversaire honorable, soit avec le loup, soit avec les agneaux (et, dans ce cas, il sera possible de définir une stratégie gagnante pour HECTOR).

```
 10 'LE LOUP ET LES AGNEAUX
 20 CLS:INPUT"AGNEAUX (0) OU LOUP (1) :";S
 30 A$="8183858714"
 40 CLSO:SPECIAL:SCALE1,1:PEN 2
 50 PRINT"    A  B  C  D  E  F  G  H"
 60 FOR I=1 TO 8:CURSOR 0,2*I:PRINTI:NEXT
 70 SCALE 3,2
 80 FOR I=1 TO 8
 90 FOR J=1 TO 8
 100 CURSOR J,I
 110 IF (I+J)/2=INT((I+J)/2) THEN PAPER 1:ELSE
PAPER 3
 120 PRINT" ";
 130 NEXT J:NEXT I
 140 SCALE 1,2:PEN 2:PAPERO
 150 CURSOR 1+3*VAL(MID$(A$,10,1)),VAL(MID$(A$,
9,1)):PRINT"X"
 160 CURSOR 1+3*VAL(MID$(A$,2,1)),VAL(MID$(A$,1
,1)):PRINT"1"
 170 CURSOR 1+3*VAL(MID$(A$,4,1)),VAL(MID$(A$,3
,1)):PRINT"2"
 180 CURSOR 1+3*VAL(MID$(A$,6,1)),VAL(MID$(A$,5
,1)):PRINT"3"
 190 CURSOR 1+3*VAL(MID$(A$,8,1)),VAL(MID$(A$,7
,1)):PRINT"4"
 200 S=ABS(S-1)
 210 SCALE 1,1:CURSOR 23,22:PRINT"
                    "
 220 CURSOR 0,22:IF S=0 THEN INPUT"JEU DU LOUP
(Q,S,W,X) :";R$:IF R$="" THEN GOTO 320
 230 IF S=1 THEN INPUT"JEU DES AGNEAUX (ND OU N
G) :";R$:IF R$="" THEN GOTO 320 :ELSE GOTO 280
 240 C=10
 250 A=VAL(RIGHT$(A$,2))+((C+1) AND (R$="X"))+(
(C-1) AND (R$="W"))
 260 A=A-((C+1) AND (R$="Q"))-((C-1) AND (R$="S
"))
 270 GOTO 300
 280 C=2*VAL(MID$(R$,1,1))
 290 A=VAL(MID$(A$,C-1,2))-(9 AND RIGHT$(R$,1)=
"D")-(11 AND RIGHT$(R$,1)="G")
 300 A$=LEFT$(A$,C-2)+RIGHT$(STR$(A),2)+RIGHT$(
A$,LEN(A$)-C)
 310 GOTO 70
 320 STANDARD
 999 STANDARD:HOME
```

Niveau 4

Nom du programme	*Thèmes et particularités*
Tirage de dés	Suffisamment souple pour être utilisé partout.
Jeu de loto	Gagnez sans problème, au prochain tirage.
La bombe	Saurez-vous la découvrir à temps ? (1 joueur).
Affichage publicitaire	Si vous ne savez pas quoi mettre dans votre vitrine.
Evasion	Il vous faudra beaucoup de chance pour réussir.
Chapeaux	Soyez très observateur, et vous réussirez (2 joueurs).
Combat aérien	Pas si facile de descendre un appareil qui bouge tout le temps.
Le jeu du missile	Une mission bien périlleuse (un seul joueur).
Le singe	Un jeu de lettres pour 2 joueurs, où il faut de bons réflexes.
Le jeu du 15	Un jeu de réflexion utilisant des chiffres (2 joueurs).
Adresse	Jeu de réflexe pour 1 joueur, très visuel.
Sous-marin	Un classique des jeux vidéo (1 joueur).
Zombies	Un peu farfelu, mais très visuel (1 joueur).
Diago	Etes-vous vraiment sûr de vos réflexes ?
Décomaque	Pour vous éviter de sécher trop longtemps sur "Nicomaque".
Chasse au sous-marin	Il vous faudra pas mal de réflexion, et beaucoup de chance.
Clavier	Apprenez à taper à la machine en vous amusant.
Le lézard d'Alice	Beaucoup de hasard, mais très spectaculaire.
Calculatrice	Un programme de simulation très réaliste.

Les techniques d'animation élémentaire

SAISIE AU VOL D'UN CARACTERE FRAPPE AU CLAVIER : INKE Y$

Le principe d'un jeu graphique, en animation, est de laisser le jeu se dérouler tout seul, d'une certaine manière, pendant un certain temps, tant qu'aucune intervention n'a été faite par le joueur.

Il faut donc disposer de la possibilité d'observer s'il y a eu ou non intervention du joueur, sans pour cela interrompre le déroulement du programme.

Dans le Basic de HECTOR, cette propriété est offerte, par la **fonction INKEY$** qui effectue une scrutation des touches du clavier pendant le temps indiqué en paramètre.

Exemple : A$=INKEYS (50)
A$ contient le caractère s'il y a eu frappe, et CHR$ (Ø) sinon.
Les codes des flèches sont :

```
        29
        ↑
8  ←         →  9
        ↓
        30
```

Il sera quelquefois nécessaire de taper plusieurs fois la touche pour qu'elle soit prise en compte.

DEPLACEMENTS D'OBJETS DE BAS EN HAUT SUR L'ECRAN : SCROLL

Il est difficile, si l'on travaille en Basic, de gérer le déplacement simultané de plusieurs objets sur l'écran (sinon on obtient vite une animation saccadée, à cause d'un temps de déplacement prohibitif).

HECTOR réalise automatiquement un **"Scrolling"** (c'est-à-dire un déplacement de l'ensemble de l'image) vers le haut, lorsque l'on écrit sur la dernière ligne en bas de l'écran.

Cette propriété a été mise à profit dans plusieurs jeux : cela permet en effet, de déplacer d'une ligne vers le haut les divers objets (textes ou graphiques) présents sur l'écran à un moment donné. La ligne du haut est bien évidemment perdue.

On peut également réaliser ce déplacement par l'introduction SCROLL.

SAISIE D'UNE CHAINE DE LONGUEUR DONNEE : INSTR$

Utilisée la plupart du temps pour saisir un caractère, cette fonction permet de ne pas taper ENTREE.

Exemple : A$=INSTR$ (1)

Dès qu'une touche sera tapée, A$ contiendra le caractère correspondant.

Tirage de dés

REFERENCES :
Difficulté de programmation : aucune.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : aucune.

LE JEU :
Le programme proposé ne réalise pas un jeu en lui-même, mais pourra être utilisé en sous-programme pour de nombreux jeux.
Il offre simplement la possibilité de lancer de 1 à 9 dés, par simple pression d'une touche. Les dés sont ensuite affichés avec un graphisme très simplifié.

LE PROGRAMME :
La ligne 6Ø saisit le caractère frappé au clavier et le transforme en nombre (si ce n'est pas possible, le programme refuse l'entrée).
Une variable D non dimensionnée reçoit la valeur indiquée par chaque dé.
Une boucle I (1ØØ-13Ø) sert au tirage (tirage en ligne 11Ø et impression en lignes 12Ø et 14Ø).

POUR JOUER :
On lance le programme par l'ordre RUN et on obtient l'affichage NOMBRE DE DES A LANCER auquel on répond en frappant un chiffre.
Le résultat est alors automatiquement affiché.
Pour arrêter, il est nécessaire de taper CNT et C.

EXTENSIONS POSSIBLES :
On pourra, bien sûr, utiliser le canevas fourni par ce programme pour la réalisation de tout jeu se servant de dés (421, YAM'S, CRAPS, etc.).

```
 10 'TIRAGE DE DES
 20 SEEDTIME(0)
 30 CLS:STANDARD:COVER1:PAPER0:PEN2
 40 HOME
 50 PEN 2:PRINT"NOMBRE DE DES A LANCER .... : "
;
 60 A$=INSTR$(1)
 70 IF VAL(A$)=0 THEN GOTO 160
 80 N=VAL(A$)
 90 CLS:PRINT:PRINT:PRINT:PRINT:PEN 3:FOR I=1 T
O N:PRINTCHR$(127);CHR$(127);CHR$(127);" ";:NEX
T:PRINT
 100 FOR I=1 TO N
 110 D=INT(RND(1,7))
 120 PEN3:PRINTCHR$(127);:PEN1:PRINTMID$(STR$(D
),2);:PEN3:PRINTCHR$(127);" ";
 130 NEXT I
 140 PRINT:PEN 3:FOR I=1 TO N:PRINTCHR$(127);CH
R$(127);CHR$(127);" ";:NEXT:PRINT
 150 GOTO40
 160 PEN2:COVER0
```

Jeu de loto

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : hasard (plusieurs joueurs).
Difficulté du jeu : facile.

LE JEU :
Il s'agit de réaliser un tirage complet de loto, c'est-à-dire d'obtenir 6 numéros, compris entre 1 et 49, plus un numéro complémentaire, tous différents.. qui pourraient être, par exemple, utilisés pour remplir votre prochaine grille de loto.

LE PROGRAMME :
On utilisera un tableau N(7), dimensionné en ligne 4Ø, pour enregistrer les résultats.
Le tirage des nombres est effectué dans une boucle I (6Ø-12Ø), une deuxième boucle J (9Ø-11Ø) étant utilisée pour contrôler qu'il n'y a pas deux tirages identiques.
Le tirage (ou le retirage) est effectué en ligne 7Ø.
La boucle I (16Ø-19Ø) affiche le résultat, et la ligne 2ØØ affiche le numéro complémentaire.
Les lignes 22Ø à 24Ø s'occupent du cas où l'on voudrait refaire une autre partie.

POUR JOUER :
Il suffit de taper RUN pour lancer le programme et obtenir l'affichage d'un premier loto.
La question VOULEZ-VOUS UN AUTRE TIRAGE (O/N) s'affiche en bas de l'écran ; on répondra O si l'on désire continuer.

EXTENSIONS POSSIBLES :
On peut imaginer un programme qui pourrait tenir compte, soit de ses propres résultats, soit de résultats entrés par un joueur, pour améliorer le tirage, en tenant compte des probabilités de sortir un numéro donné.
On peut aussi envisager un programme qui réaliserait des cartons pour le jeu de loto plus usuel, tel qu'il est encore couramment pratiqué dans le Midi de la France.

```
 10 'JEU DE LOTO
 20 SEEDTIME(0)
 30 CLS:SPECIAL:SCALE2,1
 40 DIM N(7)
 50 CLS
 60 FOR I=1 TO 7
 70 N(I)=INT(RND(1,50))
 80 IF I=1 THEN GOTO 120
 90 FOR J=1 TO I-1
 100 IF N(I)=N(J) THEN J=I:NEXT J:GOTO 70
 110 NEXT J
 120 NEXT I
 130 CURSOR0,0:SCALE 2,2:PEN 3
 140 PRINT"TIRAGE DU LOTO"
 150 PRINT"______________"
 160 PEN 1:SCALE 1,2:CURSOR 0,5:FOR I=1 TO 6
 170 PAUSE.5:BEEP:PRINTN(I);
 180 NEXT I
 190 PRINT
 200 CURSOR 0,8:PRINT"N. COMPLEMENTAIRE :";:PAU
SE 1:BEEP:PRINTN(7):PAUSE 1
 210 SCALE 1,1:CURSOR 0,22:PEN 2:SIREN
 220 PRINT"UN AUTRE TIRAGE ? (O/N) :";:A$=INSTR
$(1)
 230 HUSH
 240 IF A$="O" THEN GOTO 50
 250 STANDARD
```

La bombe

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

LE JEU :
Vous êtes censé trouver, à l'aide d'un détecteur spécialisé, un colis radioactif piégé disposé par un voyageur mal intentionné, dans une chambre d'hôtel. Seul problème, mais de taille : l'hôtel comporte un million de chambres (1ØØ*1ØØ*1ØØ).
Votre détecteur affiche, en permanence, un signal qui vous renseigne sur la proximité de la bombe, et le temps qui vous reste est affiché en permanence, décompté de 1Ø en 1Ø.
Pourrez-vous découvrir la bombe avant l'explosion ?

LE PROGRAMME :
Une variable numérique R, non dimensionnée, contient les trois coordonnées définissant la position du joueur.
La position de la chambre contenant le colis piégé est définie par un tirage aléatoire de 3 variables A, B, C, en ligne 5Ø.
Les lignes 8Ø à 1ØØ dessinent le détecteur.
Le signal de proximité est calculé en ligne 12Ø et la ligne 14Ø permet l'affichage des coordonnées de la chambre à découvrir, dans le cas où l'on a perdu.
La ligne 15Ø permet l'affichage, à la fois du signal et du temps restant à courir.
La ligne 16Ø réalise le décompte du temps.
La ligne 17Ø contrôle si l'on a trouvé ou non.
Les lignes 18Ø à 25Ø permettent d'entrer les nouvelles coordonnées.

POUR JOUER :
Après le lancement du programme par RUN, le signal et le temps (en principe 2ØØ) sont affichés.
Le joueur est alors en position x=Ø, y=Ø, z=Ø, et on devra introduire successivement les trois coordonnées x, y, z de la chambre à explorer.
On fera, par exemple, 13 puis ENTREE, 45 puis ENTREE, etc.
Le nouveau signal est alors affiché, et le temps est décrémenté de 1Ø.
On continuera ainsi, soit jusqu'au bon résultat (arrêt du jeu et fin de programme), soit jusqu'à l'apparition de 3 nombres (x,y,z) sans sortie du programme, ce qui signifie alors que l'on a perdu.

EXTENSIONS POSSIBLES :
Il sera sans doute indispensable d'améliorer le graphisme, d'ajouter une explosion (sonore et visuelle) pour symboliser l'éclatement de la bombe, d'afficher symboliquement la position dans l'hôtel, etc.

```
 10 'LA BOMBE
 20 SEEDTIME(0)
 30 CLS:SPECIAL:SCALE1,2:PEN2:PAPER0
 40 L=100:T=200
 50 A=INT(RND(0,L)):B=INT(RND(0,L)):C=INT(RND(0
,L))
 60 CLS
 70 PAPER 3
 80 CURSOR 0,2:PRINT"                  "
 90 CURSOR 0,3:PRINT"                  "
 100 CURSOR 0,4:PRINT"                  "
 110 PAPER0
 120 S=ABS((A/L+B+C*L)-(R(1)/L+R(2)+R(3)*L))
 130 S$=LEFT$(STR$(S),7):S=VAL(S$)
 140 IF T=0 THEN PAPER 3:PEN 1:CURSOR0,2:PRINT"
PERDU. SOLUTION :":PRINT:PRINTA;B;C:STANDARD:PE
N 2:PAPER 0:END
 150 PEN 1:CURSOR 2,3:PRINTS:CURSOR 12,3:PRINTT
:PEN 2
 160 SHOT 1:T=T-INT(SQR(L))
 170 IF S=0 THEN CURSOR 0,12:PRINT"BRAVO ...":F
ORI=1TO10:BELL:SHOT0:NEXT:STANDARD:END
 180 CURSOR0,7:PRINT"---> X :          "
 190 CURSOR0,8:PRINT"---> Y :          "
 200 CURSOR0,9:PRINT"---> Z :          "
 210 FOR I=1 TO 3
 220 R$=MID$(STR$(R(I)),2)
 230 CURSOR9,6+I:RECTIFY R$
 240 R(I)=VAL(R$)
 250 NEXT I
 260 GOTO 70
```

Affichage publicitaire

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : affichage publicitaire.
Difficulté du jeu : aucune.

LE JEU :
Un texte quelconque devra être introduit en début de programme. Il pourra être relativement long (4 ou 5 lignes) ; on choisira ensuite le sens et la vitesse de défilement, à la suite de quoi le texte va apparaître (une seule ligne à la fois) et commencera à défiler, à la vitesse choisie, et dans le sens choisi.

LE PROGRAMME :
Le texte sera introduit lettre par lettre aux lignes 6Ø à 9Ø.
L'affichage sera tronqué à 27 caractères au plus, grâce à la ligne 17Ø.
Le défilement vers la gauche ou la droite est préparé à la ligne 18Ø.
Le texte sera affiché en ligne 17Ø, et la durée de l'affichage (vitesse de défilement) est choisie par la ligne 19Ø.
La ligne 2ØØ permettra de changer, si besoin est, les conditions du défilement.

POUR JOUER :
On lancera le programme par RUN, et à l'affichage de la question ENTREZ VOTRE TEXTE, on introduira le texte à faire défiler, suivi de ENTREE.
On obtient ensuite l'affichage DEFILEMENT A DROITE (D) OU A GAUCHE (G) auquel on répondra en tapant, soit D, soit G, suivi de ENTREE.
Puis à la question VITESSE, on répondra en introduisant un petit nombre pour une vitesse rapide, et un grand pour une vitesse lente (par exemple, de Ø à 4Ø ou plus).
Le texte commencera à défiler dès que l'on aura frappé ENTREE.
Si l'on désire changer les conditions du défilement, il suffit d'appuyer sur la touche Ø pour retrouver les questions SENS et VITESSE. On tapera CNT et C pour interrompre le programme.

EXTENSIONS POSSIBLES :
On pourra introduire de la couleur, réaliser un affichage ondulant, faire remonter les lignes sur l'écran, etc.

```
 10 'AFFICHAGE PUBLICITAIRE
 20 CLS:STANDARD
 30 B$=""
 40 PRINT"ENTREZ VOTRE TEXTE :"
 50 PRINT
 60 A$=INSTR$(1)
 70 IF ASC(A$)=13 THEN GOTO 95
 80 B$=B$+A$
 90 PRINTA$;:GOTO 60
 95 B$=B$+" "
 96 IF LEN(B$)<27 THENB$=B$+B$:GOTO 96
 100 PRINT:PRINT:PRINT"DEFILEMENT A DROITE OU G
AUCHE (D/G):";:A$=INSTR$(1):PRINTA$
 110 PRINT:PRINT
 120 INPUT"VITESSE (DE 0 A 50) :";V
 130 V=51-V
 140 CLS:SPECIAL:SCALE 1,2
 150 T=LEN(B$)
 160 CURSOR 0,5
 170 PRINTMID$(B$,1,27)
 180 IF A$="G" THEN B$=RIGHT$(B$,T-1)+LEFT$(B$,
1) :ELSE B$=RIGHT$(B$,1)+LEFT$(B$,T-1)
 190 C$=INKEY$(V)
 200 IF C$="0" THEN STANDARD:HOME:GOTO 100
 210 GOTO 160
```

Evasion

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : solitaire.
Difficulté du jeu : facile.

LE JEU :
Votre seul problème : réussir à traverser l'écran, sans que HECTOR vous ait auparavant touché par l'un des nombreux obus explosifs qu'il va tirer sur vous.
Vous pouvez vous déplacer dans 4 directions (à l'aide des 4 flèches), et vous devez réussir à sortir du terrain par le bord droit, sans, bien sûr, avoir été touché une seule fois.

LE PROGRAMME :
La ligne 50 définit la position de départ du joueur.
La ligne 60 choisit la coordonnée horizontale de la case où HECTOR va tirer (liée à votre position), et la ligne 70 choisit la coordonnée verticale (arbitraire) de cette case.
Le déplacement du joueur est contrôlé par les lignes 140 et 150 ; sa trajectoire est tracée par la ligne 120, et sa position actuelle (sur l'écran, un #) est imprimée par la ligne 160.
La ligne 170 regarde s'il est touché ou non.
La ligne 180 effectue le contrôle de fin de partie.

POUR JOUER :
On commence l'exécution du programme par RUN.
Un premier tir a lieu, et ce n'est qu'après que le joueur apparaît, représenté par un #.
Les 4 touches de direction (←, ↓, ↑, →) servent à déplacer le joueur.
S'il est touché, on obtient, sur l'écran, le message PERDU et la partie s'arrête.
S'il réussit à sortir par la droite de l'écran sans être touché, on obtient le message BRAVO, et la partie est également terminée.
On peut terminer à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
La couleur et un graphisme plus soigné amélioreront considérablement l'attrait du jeu. On pourra aussi rendre le tir des bombes plus aléatoire, introduire des obstacles à contourner, ajouter des contrôles, faire participer plusieurs joueurs, etc.

```
 10 'EVASION
 20 SEEDTIME(0)
 30 WIPE:LITTLE
 40 X=3
 50 Y=12
 60 H=6*X+INT(RND(3,16))
 70 K=INT(RND(6,220))
 80 IF X<3 THEN GOTO 120
 90 FOR J=1 TO RND(3,12)
 100 INK1:PLOT H+RND(-2,5),K+RND(-2,5)
 110 NEXT J
 120 PEN3:BEEP:BEEP:CURSOR X*6+9,226-Y*10:PRINT
"."
 130 SHOT1:A$=INKEY$(30)
 140 X=X-(A$=CHR$(9))+(A$=CHR$(8))
 150 Y=Y-(A$=CHR$(30))+(A$=CHR$(29))
 160 CURSOR X*6+9,226-Y*10:PRINT"#"
 170 IF X>3 AND (ABS(H-6*X-12)+ABS(K-226+10*Y))
<=10) THEN LASER:LASER:PRINT:PRINT:PRINT"===  P
ERDU ...":STANDARD:PEN2:END
 180 IF X<36 THEN GOTO 60
```

Chapeaux

REFERENCES :
Difficulté de programmation : assez facile.
Catégorie de jeu : jeu de réflexe, pour deux joueurs.
Difficulté du jeu : facile.

LE JEU :
L'idée en est très simple : deux personnages se déplacent l'un vers l'autre sur l'écran et se croisent. Au moment où ils se croisent, il est possible qu'ils échangent leurs chapeaux. Il s'agit d'être le premier à dire si oui ou non il y a eu échange de chapeaux lors du croisement.

LE PROGRAMME :
La ligne 2Ø contient le dessin des deux chapeaux (dans deux variables alphanumériques B$ et C$).
La ligne 3Ø contient, dans une autre variable A$, les symboles qui permettront de dessiner les personnages.
La ligne 5Ø permet, s'il y a lieu, l'échange des chapeaux.
La variable T de la ligne 6Ø sert à repérer s'il y a eu échange ou non.
Le dessin des deux personnages est réalisé par une boucle I (11Ø-14Ø), et leur déplacement est réalisé par la ligne 15Ø.
Le test, permettant ou non le changement des couvre-chefs, est effectué en ligne 16Ø.
Les lignes 17Ø à 26Ø déterminent si le premier joueur ayant appuyé sur une touche a gagné ou perdu.

POUR JOUER :
Lancer le programme par RUN.
Les personnages commencent immédiatement à se déplacer l'un vers l'autre.
Le joueur de gauche dispose des touches 1 et 2 et celui de droite, des touches 9 et Ø.
S'il y a eu changement de chapeau, les touches à appuyer seront 1 (à gauche) et Ø (à droite).
S'il n'y a pas eu de changement, il faudra, pour gagner, être le premier à appuyer sur 2 (à gauche) et sur 9 (à droite).
L'affichage indiquera alors BRAVO GAUCHE ou BRAVO DROITE, ou encore, PERDU GAUCHE ou PERDU DROITE.

EXTENSIONS POSSIBLES :
Il sera possible de prévoir un affichage couleur, avec une animation des personnages.
On pourra aussi compliquer le jeu à plaisir avec plusieurs objets échangés de façon aléatoire (il faudrait alors en trouver la liste complète).

```
 10 'CHAPEAUX
 20 B$=" ==_":C$=" _=="
 30 A$=" ###  ###   #    #  ###################
# # #  # #  # # ## ##"
 40 X=0:Y=22:T=1
 50 D$=C$:C$=B$:B$=D$
 60 T=ABS(T-1)
 70 CLS:SPECIAL:SCALE 1,2
 71 SCALE1,1
 80 IF X>22 OR Y<0 THEN GOTO 170
 90 CURSOR X,0:PRINT" ";C$;" "
 100 CURSORY,0:PRINT" ";D$;" "
 110 FOR I=0 TO 11
 120 CURSOR X,I+2:PRINT" ";MID$(A$,I*5+1,5);" "

 130 CURSOR Y,I+2:PRINT" ";MID$(A$,I*5+1,5);" "

 140 NEXT I
 150 X=X+1:Y=Y-1
 160  IF X=Y AND RND(0,1)<.5 THEN GOTO 50
 170 BEEP:K$=INKEY$(20):IF ASC(K$)=0 THEN GOTO
80
 180 K=VAL(K$)
 190 IF K<>1 AND K<>2 AND K<>9 AND K<>0 THEN GO
TO 170
 200 IF X<=Y THEN GOTO 220
 210 IF (K=1 AND T=1) OR (K=2 AND T=0) THEN PRI
NT:PRINT"=== BRAVO GAUCHE ...":GOTO 250
 220 IF (K=0 AND T=1) OR (K=9 AND T=0) THEN PRI
NT:PRINT"=== BRAVO DROITE ...":GOTO 250
 230 IF K=2 OR K=1 THEN PRINT:PRINT"=== PERDU G
AUCHE !":GOTO 250
 240 IF K=9 OR K=0 THEN PRINT:PRINT"=== PERDU D
ROITE !"
 250 LASER
 260 STANDARD
```

Combat aérien

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexe (1 joueur).
Difficulté du jeu : difficile.

LE JEU :
Au centre de l'écran, se trouve le viseur de votre appareil, et l'avion ennemi se déplace devant vous, d'une façon qui pourra vous sembler au début incohérente.
Vous pouvez vous déplacer dans toutes les directions, grâce aux flèches de mouvement, et vous allez essayer d'amener l'appareil ennemi au centre de votre viseur.
Vous tirez sur lui en frappant Ø, et vous voyez vos projectiles qui convergent sur lui. Mais il continue à se déplacer, même pendant votre tir, et il n'est pas du tout évident que vous réussirez à l'avoir.
La partie continuera à se dérouler jusqu'à ce que vous l'ayez touché (affichage de GAGNE) ou qu'il soit sorti de votre champ de vision.

LE PROGRAMME :
La ligne 7Ø dessine l'appareil ennemi dont le déplacement aléatoire sera réalisé par les lignes 9Ø et 1ØØ.
Les lignes 12Ø et 13Ø vous permettent le pilotage de votre appareil.
La ligne 17Ø dessine le viseur de votre appareil.
Le sous-programme, gérant le tir, va de la ligne 2ØØ à la ligne 25Ø et le test de fin de partie est effectué en ligne 24Ø.
La ligne 18Ø regarde si vous avez tiré ou non.

POUR JOUER :
Dès que le programme a été lancé par l'ordre RUN, vous devez manœuvrer pour essayer de faire en sorte que l'appareil ennemi soit cadré dans votre viseur.
Pour cela, vous disposez des flèches de déplacement (touche ↑, ↓, →, ←) et vous pouvez tirer à tout moment en pressant la touche Ø.
L'appareil ennemi peut continuer à se déplacer de manière aléatoire, même lorsque vous avez tiré.
La partie se terminera avec l'affichage de GAGNE, si vous touchez l'avion ennemi.
On peut aussi arrêter à tout moment en frappant CNT et C.

EXTENSIONS POSSIBLES :
Il sera assez facile de faire un affichage au graphisme plus soigné, utilisant la couleur. On pourrait envisager de stabiliser la position de l'avion ennemi pendant le tir (il suffit de modifier la ligne 18Ø).
On peut aussi introduire plusieurs avions, compter les points ou le temps, tenir compte de sa propre réserve de carburant, etc.

```
 10 'COMBAT AERIEN
 20 SEEDTIME(0)
 30 WIPE:SPECIAL:SCALE 1,1:PEN2
 40 CLS
 50 A=5:B=15
 60 N=17
 70 CURSOR B,A:PEN1:PRINT"#-#"
 80 U=A:V=B
 90 A=A+2*INT(RND(0,2))-1
 100 B=B+2*INT(RND(0,2))-1
 110 A$=INKEY$(20)
 120 B=B-2*(A$=CHR$(9))+2*(A$=CHR$(8))
 130 A=A-2*(A$=CHR$(30))+2*(A$=CHR$(29))
 140 IF B<0 OR B>25 THEN B=ABS(B-25)
 150 IF A<3 OR A>22 THEN A=ABS(A-22)
 160 CURSOR V,U:PRINT"    "
 170 CURSOR 12,3:PEN3:PRINT"* *":CURSOR12,4:PRI
NT" X ":CURSOR 12,5:PRINT"* *"
 180 IF A$="O" OR N<17 THEN GOSUB 200
 190 GOTO 70
 200 PEN2:CURSOR 17-N,N:PRINT"*":CURSOR N+9,N:P
RINT"*"
 205 BEEP
 210 CURSOR 17-N,N:PRINT" ":CURSOR N+9,N:PRINT"
 "
 220 N=N-1
 230 IF N>3 THEN RETURN
 240 IF A<3 OR A>5 OR B<12 OR B>14 THEN BELL:GO
TO 60
 250 LASER:PRINT:PRINT:PRINT"===  GAGNE"
 260 STANDARD
```

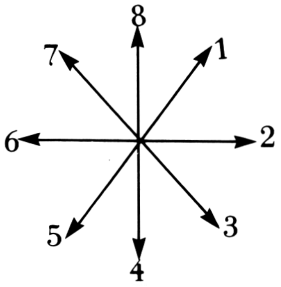
Le jeu du missile

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : contre HECTOR.
Difficulté du jeu : moyenne.

LE JEU :
HECTOR est en alerte : vous êtes aux commandes d'un avion chargé d'une mission d'espionnage sur son territoire ; votre but est de survoler une base ennemie (représentée par un H agrandi), et de revenir sain et sauf sur votre base ; mais, dès que HECTOR vous aura repéré, il va envoyer à votre poursuite, un missile très perfectionné qui n'aura de cesse de vous détruire. Votre avion peut se déplacer dans toutes les directions, grâce aux touches de 1 à 8 ; réussirez-vous à remplir votre mission ?

LE PROGRAMME :
La ligne 5Ø dessine la base ennemie à survoler (sous forme d'un H agrandi).
La ligne 6Ø dessine votre base de départ.
Les lignes 8Ø et 1ØØ choisissent l'instant de départ du missile.
La ligne 12Ø dessine votre avion.
Les lignes 14Ø et 15Ø définissent son mouvement suivant 8 directions, en utilisant les sous-programmes en lignes 1ØØØ, 2ØØØ, 3ØØØ et 4ØØØ.
La ligne 16Ø assure le départ du missile qui est dessiné par la ligne 17Ø et dont le déplacement est géré par les lignes 18Ø et 19Ø.
Les lignes 2ØØ et 21Ø contrôlent la fin de partie.

POUR JOUER :
Lancer le programme par RUN.
Le terrain de jeu s'affiche et vous pouvez déplacer votre appareil en utilisant les touches de 1 à 8, avec la convention suivante :



Vous devez lui faire contourner le territoire ennemi (si vous, ou le missile, touchez ce territoire, une deuxième partie commence).
Si le missile vous détruit, vous obtenez un message TOUCHE, et la partie s'arrête.
Taper CNT et C pour arrêter la partie.

EXTENSIONS POSSIBLES :
Il sera sans doute nécessaire de rajouter divers contrôles dans le programme (l'avion a-t-il survolé la base ennemie, est-il revenu à sa propre base...). On peut aussi prévoir divers niveaux de jeu en faisant, par exemple, varier la vitesse du missile, prévoir l'utilisation de leurres par l'avion, l'explosion du missile au bout d'un certain temps, etc.

```
 10 'LE JEU DU MISSILE
 20 SEEDTIME(0)
 30 WIPE:M=0
 40 SPECIAL:SCALE1,1:PEN2
 50 SCALE 2,2:CURSOR6,2:PRINT"H"
 60 CURSOR 0,10:PAPER 2:PRINT"  ":PAPER 0
 70 U=15:X=18
 80 S=INT(RND(6,16))
 90 Y=1
 100 S=S-1
 110 V=27
 120 PEN 3:SCALE1,1:CURSOR Y,X:PRINT"Y"
 130 A$=INSTR$(1)
 140 F=-2*(VAL(A$)<5)-1
 150 ON VAL(A$)+2*(F-1) GOSUB 1000,2000,3000,40
00
 160 IF S<>0 THEN GOTO 100
 170 CURSOR V,U:PEN 1:PRINT"<"
 180 U=U-SGN(U-X)
 190 V=V-SGN(V-Y)
 200 IF U=X AND V=Y THEN CURSOR 0,24:PRINT"TOUC
HE ....":GOTO 250
 210 IF (X=4 OR X=5) AND (Y=12 OR Y=13) THEN GO
TO 40
 220 IF X<=4 AND Y>=13 AND M=0 THEN M=1:BELL
 230 IF M=1 AND X=19 AND Y<=4 THEN CURSOR 0,24:
PRINT"MISSION ACCOMPLIE":GOTO 250
 240 GOTO 120
 250 LASER:PEN 2:STANDARD
 260 END
 1000 X=X-F
 2000 Y=Y+F
 2010 RETURN
 3000 GOSUB2000
 4000 X=X+F
 4010 RETURN
```

Le singe

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : deux joueurs.
Difficulté du jeu : difficile.

LE JEU :
Il s'inspire de l'idée déjà ancienne du singe qui, frappant au hasard sur une machine à écrire, a des chances non nulles de finir par écrire un texte signifiant.
Le rôle du singe est, ici, tenu par HECTOR qui va afficher des groupes de 3 à 7 lettres pendant un temps relativement court.
Si l'un des joueurs estime qu'il peut réussir à faire un mot avec les lettres proposées, il doit taper rapidement sur sa touche de jeu (1 ou Ø).
Le mot devra alors être accepté ou refusé par l'adversaire qui répondra JUSTE ou FAUX.
Chaque mot juste rapporte 1 point, chaque mot faux enlève 1 point.

LE PROGRAMME :
La ligne 4Ø contient une liste V$ des voyelles courantes, et une liste C$ des consonnes courantes.
Il sera possible de modifier ces deux listes pour augmenter, par exemple, la fréquence d'apparition de certaines lettres (on pourra ainsi mettre deux E dans V$ ou deux R dans C$, etc.).
La ligne 5Ø définit la longueur du mot (en principe, de 3 à 7 lettres, mais on peut également modifier cette longueur).
La boucle I (8Ø-13Ø) permet la confection d'un "mot" en prenant une lettre, soit dans l'alphabet complet, soit dans la liste C$ (ligne 1ØØ), soit dans la liste V$ (ligne 11Ø).
La ligne 15Ø temporise l'apparition du résultat.
La ligne 17Ø gère la fin de partie, avec affichage du score.
La ligne 18Ø regarde si l'un des joueurs a réagi.
Enfin, les lignes 2ØØ et 21Ø comptabilisent les points de chaque joueur.

POUR JOUER :
On lance le programme en tapant RUN.
La première série de lettres est alors affichée sur l'écran. Le joueur de gauche devra taper 1 s'il voit qu'il peut réaliser un mot, celui de droite devra taper Ø.
Si rien n'a été frappé, une autre série de lettres sera affichée après peu de temps.
Si l'un des joueurs a frappé Ø ou 1, la question (J)USTE ou (F)AUX est alors affichée. On répond J si le mot est correct, F si le mot est incorrect, suivi de ENTREE.
Les points sont alors calculés par HECTOR, et la série de lettres suivante est alors affichée.
Pour sortir du programme, il faudra taper P au lieu de Ø ou 1 ; le score final est alors affiché.

EXTENSIONS POSSIBLES :
On pourra modifier la longueur des mots, la fréquence d'apparition des lettres, le décompte des points, la règle du jeu, etc.

```
 10 'LE SINGE
 20 SEEDTIME(0)
 30 WIPE:SPECIAL:SCALE1,1:PEN2
 40 A=0:B=0:V$="AEIOU":C$="BCDGLMNPRSTV"
 50 X=INT(RND(3,8))
 60 M$=""
 70 CLS
 80 FOR I=1 TO X
 90 E$=CHR$(INT(RND(65,91)))
 100 IF RND(0,1)<.4 THEN E$=MID$(C$,INT(RND(1,L
EN(C$)+1)),1)
 110 IF RND(0,1)<.5 THEN E$=MID$(V$,INT(RND(1,L
EN(C$)+1)),1)
 120 M$=M$+E$
 130 NEXT I
 140 SCALE 2,2:CURSOR2,3:PEN1:PRINTM$:BEEP
 150 A$=INKEY$(100)
 160 SCALE 1,1:PEN2
 170 IF A$="P" THEN GOTO 230
 180 IF A$<>"0" AND A$<>"1" THEN GOTO 50
 190 PRINT:INPUT"(J)USTE OU (F)AUX :";R$
 200 A=A+(R$="J")*(A$="0")-(R$="F")*(A$="0")
 210 B=B+(R$="J")*(A$="1")-(R$="F")*(A$="1")
 220 GOTO 50
 230 CURSOR 0,12:PRINT"SCORE DE A :";A
 240 PRINT:PRINT"SCORE DE B :";B
 250 STANDARD
 260 PEN2
```

Le jeu du 15

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :
C'est un jeu pour deux joueurs, assez ancien, qui était autrefois pratiqué dans les foires. On dispose d'un tapis de jeu de 9 cases, et chacun mise à son tour un jeton de sa couleur, sur une case disponible.
Le premier des deux joueurs qui, avec exactement 3 jetons, arrive à faire un total de 15, a gagné.

LE PROGRAMME :
Les lignes 5Ø à 85 dessinent le terrain de jeu, et les lignes 9Ø à 1ØØ s'occupent de la gestion du tour des joueurs (numéro 1 ou 2 pour chaque joueur, et jeton O ou X correspondant).
La ligne 11Ø attend le coup joué.
La boucle I (12Ø-14Ø) vérifie la validité du coup.
La ligne 15Ø teste la fin de partie, en cas de match nul, et la ligne 16Ø refuse les coups non conformes (avec retour à la situation initiale, ligne 9Ø).
Les lignes 17Ø et 18Ø s'occupent de l'affichage du coup, et la ligne 19Ø l'enregistre dans une variable R$. La partie la plus intéressante du programme, qui pourrait servir à définir une stratégie machine, va de la ligne 21Ø à la ligne 3ØØ : on utilise 3 boucles emboîtées pour vérifier si l'un des joueurs a obtenu, avec 3 pions, un total de 15. Le changement de joueur s'effectue en ligne 31Ø.

POUR JOUER :
Lancer le jeu en tapant RUN.
Le terrain de jeu s'affiche, puis la question JOUEUR 1 : X, et le joueur qui a choisi les X devra donner le numéro de la case où il désire jouer si elle est libre, son coup y sera enregistré.
On obtient, ensuite, l'affichage JOUEUR 2 : O et le deuxième joueur procède de la même façon. Si l'un des joueurs gagne (total de 15 sur 3 coups), il obtient l'affichage BRAVO et la partie s'arrête.
En cas de match nul, la machine affiche C'EST FINI.
On fera une autre partie en tapant RUN.

EXTENSIONS POSSIBLES :
Le plus intéressant sera sans doute d'essayer de définir une stratégie machine (en essayant d'éviter que la machine ne joue toujours le 5 lorsqu'elle doit commencer).
On peut aussi, bien sûr, améliorer le graphisme et introduire des effets sonores pour les différents messages.

```
 10 'LE JEU DU 15
 20 PAPERO:WIPE:SPECIAL:SCALE1,1:PEN2
 30 F=0:R$="123456789"
 40 O$="":X$=""
 50 PAPER1:CURSOR 3,3:PRINT"
"
 60 CURSOR 3,6:PRINT"                    "
 70 SCALE 1,2:FOR I=3 TO 21 STEP 2
 80 CURSOR I,2:PRINT" ":NEXT I
 85 FOR I=1 TO 9:CURSORI*2+1,4:PAPERO:PEN2:PRIN
TI:NEXT
 90 SCALE 1,1:CURSOR 1,18:PRINT"A VOUS, JOUEUR"
;F+1;": ";
 100 PAPER 3:PEN 0:IF F=1 THEN PRINT"O":ELSE PR
INT"X"
 110 A$=INSTR$(1):IF A$<"1" OR A$>"9" THEN BEEP
:GOTO 110
 120 FOR I=1 TO LEN(R$)
 130 IF A$=MID$(R$,I,1) THEN X=I:I=LEN(R$):NEXT
I:GOTO170
 140 NEXT I:BEEP
 150 IF VAL(R$)=0 THEN SCALE1,1:PAPERO:PEN2:CUR
SOR 1,18:PRINT"===  C'EST FINI ......       ":G
OTO 350
 160 PEN2:PAPERO:GOTO 90
 170 SCALE1,2:IF F=1 THEN O$=O$+A$:CURSOR2*X+2,
2:PRINT"O"
 180 IF F=0 THEN X$=X$+A$:CURSOR2*X+2,2:PRINT"X
"
 190 R$=LEFT$(R$,X-1)+"O"+MID$(R$,X+1)
 200 IF F=1 THEN T$=O$:ELSE T$=X$
 210 IF LEN(T$)<3 THEN GOTO 310
 220 FOR I=1 TO 3
 230 IF I=LEN(T$) THEN I=3:GOTO 300
 240 FOR J=I+1 TO LEN(T$)
 250 IF J=LEN(T$) THEN I=3:GOTO 290
 260 FOR K=J+1 TO LEN(T$)
 270 IF VAL(MID$(T$,I,1))+VAL(MID$(T$,J,1))+VAL
(MID$(T$,K,1))=15 THEN GOTO 330
 280 NEXT K
 290 NEXT J
 300 NEXT I
 310 F=ABS(F-1)
 320 GOTO 150
 330 CURSOR1,18:PAPERO:PEN2:SCALE1,1
 340 PRINT"BRAVO, JOUEUR ";:IF F=1 THEN PRINT"O
 ...     ":ELSE PRINT"X ...     "
 350 STANDARD:PEN2:PAPERO
```

Adresse

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexe (pour 1 joueur).
Difficulté du jeu : difficile.

LE JEU :
Des chiffres, entre Ø et 9, vont apparaître au hasard, en divers emplacements de l'écran. Ils restent affichés pendant un temps très court. Le joueur doit réussir à appuyer sur la touche numérique correspondante, au moment où le chiffre est affiché.
Le chiffre tapé par le joueur sera alors affiché en grand format sur la gauche de l'écran ; et chaque réponse exacte rapporte 1Ø points. La durée de la partie est aléatoire et le score sera affiché en fin de partie.

LE PROGRAMME :
Les lignes 5Ø et 6Ø contiennent, dans deux variables alphanumériques C$ et B$, les références qui seront utilisées pour dessiner les chiffres en grand format.
L'emplacement du chiffre tiré par HECTOR est choisi aux lignes 7Ø et 8Ø, et le chiffre est tiré en ligne 9Ø.
La ligne 1ØØ réalise l'impression, et la ligne 11Ø détermine la durée d'affichage. La ligne 13Ø définit la durée du jeu, et la fin de partie est traitée par la ligne 15Ø.
La boucle I (2ØØ-23Ø) réalise le dessin grand format du chiffre frappé par le joueur.
Le contrôle de l'exactitude du résultat est effectué par la ligne 24Ø.

POUR JOUER :
Le programme sera lancé par RUN, puis un premier chiffre apparaît, après une courte pause, vers le milieu de l'écran. Le joueur devra réagir suffisamment vite, tant qu'il reste à l'affichage, et le retaper. Le chiffre qu'il a tapé se dessinera alors en haut à gauche de l'écran, en grand format.
Si le résultat est exact, le joueur marquera 1Ø points.
Le temps de déroulement de la partie sera fonction des chiffres qui apparaîtront à l'affichage.
En fin de partie, le score sera affiché. Il sera possible de s'arrêter à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
On peut envisager de modifier le jeu pour faire participer plusieurs joueurs, en modifiant éventuellement l'affichage et en codant les touches du clavier mises à la disposition de chacun des joueurs.
On pourra aussi enlever des points à chaque résultat faux, modifier les règles de calcul du score et la durée du jeu, etc.

```
 10 'ADRESSE
 20 SEEDTIME(0)
 30 T=100:S=0:C$=CHR$(127)
 40 CLS:STANDARD:COVER 1
 50 C$=C$+C$+C$+C$+"     "+C$+C$+" "+C$
 60 B$="14441333331312113131441331213112141 1333
31414114131"
 70 X=INT(RND(70,220))
 80 Y=INT(RND(100,225))
 90 N=INT(RND(0,10))
 100 PEN 1:CURSOR Y,X:PRINTN
 110 A$=INKEY$(RND(100,170))
 120 IF ASC(A$)<>0 THEN GOTO 190
 130 T=T-N
 140 CURSOR Y,X:PRINT"   "
 150 IF T>=0 THEN GOTO 70
 160 CLS:PEN2
 170 PRINT:PRINT:PRINT"VOTRE SCORE EST DE";S;"S
UR 100."
 180 COVER 0:END
 190 U=VAL(A$)
 200 PEN 3:FOR I=1 TO 5
 210 A=3*VAL(MID$(B$,5*U+I,1))-2
 220 CURSOR 21,226-I*10:PRINTMID$(C$,A,3)
 230 NEXT I
 240 IF U=N THEN S=S+10
 250 GOTO 130
```

Sous-marin

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexe (1 seul joueur).
Difficulté du jeu : facile.

LE JEU :
Vous manœuvrez en surface un chasseur de sous-marins, et des sous-marins passent très régulièrement en dessous de votre unité, à différentes profondeurs.
Vous pouvez, à tout moment, larguer des grenades sous-marines et vous pouvez même envisager de toucher plusieurs fois le même sous-marin.
Votre score sera affiché en permanence, sur la gauche de l'écran (rapport nombre de coups au but sur nombre de sous-marins apparus).

LE PROGRAMME :
La ligne 7Ø permet à HECTOR de choisir la profondeur du sous-marin.
La ligne 9Ø dessine votre bâtiment.
Les lignes 1ØØ et 11Ø affichent le score et dessinent la surface de l'eau.
La ligne 12Ø dessine le sous-marin.
Le tir est effectué par un programme spécifique placé aux lignes 2ØØ à 24Ø.
Le déplacement (très rapide) de votre navire est réalisé par les lignes 17Ø et 18Ø.
Le test, permettant de savoir si l'on a touché ou non, est effectué en ligne 14Ø.

POUR JOUER :
Taper RUN pour commencer la partie.
Le bâtiment sera déplacé grâce aux flèches droite et gauche.
La touche Ø autorise le tir.
Le navire, se déplaçant trois fois plus vite que les sous-marins, il sera possible de toucher plusieurs fois le même bâtiment.
Pour terminer la partie, il conviendra de faire CNT et C ou d'attendre 99 sous-marins.

EXTENSIONS POSSIBLES :
Il sera assez élémentaire d'améliorer la qualité graphique du jeu, d'utiliser la couleur et les sons, de faire apparaître plusieurs sous-marins se déplaçant à des vitesses différentes, éventuellement même, de permettre aux sous-marins de tirer...

```
 10 'SOUS-MARIN
 20 SEEDTIME(0)
 30 P=0
 40 C=15:S=5
 50 CLS:SPECIAL:SCALE1,1:PAPER3:PRINTSPC(85)
 60 PAPER0:CURSOR0,S:PRINT"     "
 70 T=22:S=INT(RND(5,12)):P=P+.01
 80 X=1
 90 CURSOR C,2:PAPER3:PEN1:PRINT"   <==>   "
 100 PEN0:CURSOR 0,3:PRINT"####################
########":PEN2
 110 PAPER 0:CURSOR 0,18:PRINT"SCORE :";P
 120 CURSOR T,S:PRINT"-=- "
 130 IF X=1 THEN Y=C+2*X
 140 IF S=X AND (T+1=Y OR T+2=Y) THEN P=P+1:CUR
SOR T,S:PRINT"***"
 150 T=T-1:IF P=99 THEN GOTO 250
 160 IF T<0 THEN GOTO 60
 170 A$=INKEY$(80)
 180 C=C+3*(A$=CHR$(9))*(C<17)-3*(A$=CHR$(8))*(
C>0)
 190 IF A$<>"O" AND X=1 THEN GOTO 90
 195 SOUND2,14
 200 CURSOR Y,X:IF X=1 THEN PAPER 3:ELSE PAPER
0
 210 PRINT" ":X=X+1:PAPER 0
 220 IF X>S THEN HUSH: GOTO 80
 230 CURSOR Y,X:PRINT"O"
 240 GOTO 90
 250 CURSOR0,21:PRINT"===  C'EST FINI ..."
 260 STANDARD:PEN2:PAPER0:END
```

Zombies

REFERENCES :
Difficulté de programmation : assez difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez facile.

LE JEU :
Les ZOMBIES sont des êtres malfaisants qui vont essayer de vous détruire, mais vous allez, vous aussi, essayer de les utiliser à votre profit, pour débarrasser votre espace vital des trous noirs qui l'obscurcissent. Sur l'écran, est tracé un cadre qui définit votre espace vital. Dans ce cadre, vont apparaître de façon aléatoire, les ZOMBIES (ce sont des Z inversés), les trous noirs (des carrés noirs) et le joueur (un O). Les ZOMBIES convergent vers le joueur pour essayer de le capturer, mais celui-ci se déplace plus rapidement qu'eux, et il va essayer de leur faire avaler tous les trous noirs, sans se faire prendre.
La partie se termine, soit avec la disparition du joueur, soit par l'élimination de tous les ZOMBIES.

LE PROGRAMME :
La position des ZOMBIES est définie par un tableau Z non dimensionné (c'est inutile en dessous de 10).
Les lignes 50 à 120 (boucle I) dessinent le cadre, les ZOMBIES et les trous noirs.
La ligne 130 définit la position initiale du joueur.
Une deuxième boucle I (150-240) réalise le déplacement du joueur.
Les lignes 290 à 340 contrôlent le déplacement du joueur.

POUR JOUER :
Après avoir lancé l'exécution par RUN, le joueur devra se déplacer en utilisant les 4 flèches et en veillant à ne pas sortir des limites du terrain.
Il devra manœuvrer de façon à faire venir les ZOMBIES sur les trous noirs, sans se faire lui-même coincer.
Pour arrêter la partie avant la fin, il suffira de taper CNT et C.

EXTENSIONS POSSIBLES :
Il serait intéressant d'agrandir le terrain de jeu, d'augmenter le nombre de ZOMBIES ou de trous noirs, d'améliorer les contrôles, de placer le joueur de façon plus aléatoire, éventuellement de modifier la règle.
On pourra aussi, bien sûr, améliorer la qualité du dessin et introduire la couleur.

```
 10 'ZOMBIES
 20 SEEDTIME(0)
 30 CLS:STANDARD
 40 COVER1:PAPER1
 50 FOR I=0 TO 21
 60 PEN2:CURSOR I*6+15,226:PRINTCHR$(127)
 70 CURSOR 0,226-I*10:PRINTCHR$(127);:CURSOR 14
7,226-I*10:PRINTCHR$(127);
 80 CURSOR I*6+15,16:PRINTCHR$(127);
 90 IF RND(0,1)<.5 THEN PAPER1:PEN0:CURSOR I*6+
9,226-INT(RND(1,21))*10:PRINT"#"
 100 S=INT(RND(1,15))
 110 Z(INT(I/3)+1)=S+5
 120 NEXT I
 130 K=INT(RND(1,19))
 140 P=0
 150 FOR I=1 TO 4
 160 P=P+Z(I)
 170 IF Z(I)=0 THEN GOTO 240
 180 PAPER 0:CURSOR Z(I+4)*6+9,226-Z(I)*10:PRIN
T" "
 190 Z(I)=Z(I)-SGN(Z(I)-S)
 200 Z(I+4)=Z(I+4)-SGN(Z(I+4)-K)
 210 IF POINT(Z(I+4)*6+9,226-Z(I)*10)=1 THEN Z(
I)=0:GOTO 230
 215 TONE50,50
 220 CURSORZ(I+4)*6+9,226-Z(I)*10:PAPER0:PEN2:P
RINT"Z"
 230 IF Z(I)=S AND Z(I+4)=K THEN PRINT:PRINT"PE
RDU...":END
 240 NEXT I
 250 IF P=0 THEN PRINT:PRINT"BRAVO ...":END
 260 CURSOR K*6+9,226-S*10:PRINT"O"
 265 BEEP
 270 A$=INKEY$(200)
 280 CURSOR K*6+9,226-S*10
 290 PRINT" "
 300 K=K-2*((A$=CHR$(9))-(A$=CHR$(8)))
 310 IF K<=0 OR K>=23 THEN PRINT"DOMMAGE ...":E
ND
 320 S=S-2*((A$=CHR$(30))-(A$=CHR$(29)))
 330 IF S<=0 OR S>=21 THEN K=25:GOTO310
 340 GOTO 140
```

Diago

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexe (1 seul joueur).
Difficulté du jeu : difficile.

LE JEU :
On lance le programme par RUN et on obtient un premier affichage qui vous renseigne sur la performance à réaliser (par exemple, 5 points noirs ou 17 points noirs...), puis le temps est affiché et décompté, à partir de 5 secondes. Le jeu commence lorsque l'on arrive à Ø seconde. Des ronds apparaissent alors en haut et à gauche de l'écran, formant une ligne diagonale, et il vous faudra appuyer sur la touche Ø pour arrêter l'affichage des points dès que vous estimez que le nombre de points demandé au début est apparu à l'écran.
On obtient alors un affichage BRAVO ou PERDU selon le résultat obtenu.

LE PROGRAMME :
Le nombre de points noirs sera choisi au hasard, en ligne 5Ø.
Le temps est décompté par les lignes 6Ø à 12Ø.
Les points sont imprimés et la ligne diagonale est fabriquée par les lignes 13Ø et 175.
La pression de la touche Ø est détectée en ligne 18Ø, et la ligne 19Ø contrôle la fin de partie (aucun réflexe).
Les affichages de fin sont exécutés par les lignes 19Ø à 22Ø.

POUR JOUER :
On lancera l'exécution par RUN.
On obtient alors l'affichage de la phrase LE BON REFLEXE EST A... qui indique le nombre de points gagnants.
Le temps sera ensuite décompté sous la forme ATTENTION : PLUS QUE ... SECONDES qui indique le temps restant avant l'apparition de la diagonale.
A l'affichage de Ø, la diagonale apparaît et commence à se déplacer sur l'écran.
Si l'on estime que le nombre de points réclamé a été atteint, on devra presser la touche Ø.
L'affichage indique alors BRAVO si l'on a gagné, ou PERDU dans le cas contraire ; puis la phrase AUTRE ESSAI (O/N) est affichée, répondre O pour faire une autre partie.

EXTENSIONS POSSIBLES :
On pourra envisager une adaptation du jeu à deux joueurs ou plus, en laissant la diagonale se former complètement et en testant la coïncidence lors de la pression de la touche affectée au joueur.
On peut aussi réaliser un graphisme plus soigné et modifier le principe du déplacement des objets sur l'écran (déplacement horizontal, vertical ou apparition au hasard...).

```
 10 'DIAGO
 20 SEEDTIME(0)
 30 CLS
 40 STANDARD:COVER1
 50 X=INT(RND(3,21))
 60 S=5
 70 PRINT"LE BON REFLEXE EST A";X
 80 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT"ATTENTI
ON. PLUS QUE";S;"SECONDES..."
 90 BEEP:PAUSE1:S=S-1:CURSOR123,166:PRINTS
 100 IF S<>0 THEN GOTO 90
 110 CLS
 120 U=1
 130 PRINTTAB(U);"O"
 145 TONE250-3*U,2000/(250-3/U)
 150 A$=INKEY$(30)
 160 IF A$="O" THEN GOTO 190
 170 IF U>22 THEN PRINT:PRINT"PERDU ...":GOTO21
0
 175 U=U+1
 180 GOTO130
 190 IF U<>X THEN U=23:GOTO 170
 200  PRINT:PRINT"BRAVO ..."
 210 PRINT:PRINT"AUTRE ESSAI ? (O/N) :"
 220 A$=INSTR$(1):IF A$="O" THEN GOTO 30
```

Décomaque

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : solution du programme Nicomaque.
Difficulté du jeu : aucune.

LE JEU :
C'est une réplique au programme Nicomaque, proposé dans les pages précédentes ; vous entrez les restes successivement dans les divisions par 3, 5 et 7 d'un entier compris entre 1 et 100, et HECTOR calcule et affiche cet entier. Le procédé retenu peut d'ailleurs être généralisé (à vous de trouver comment).

LE PROGRAMME :
Les lignes 30 à 50 affichent quelques explications.
La ligne 90 attend que vous frappiez un reste, qui sera refusé s'il est incompatible avec la division par le nombre affiché.
La ligne 100 emmagasine les restes dans une variable non dimensionnée A (S).
La ligne 130 réaffiche les restes entrés.
Les boucles I (140-180) et J (150-170) dessinent des petits carrés selon les restes introduits.
Les lignes 210 à 230 gèrent la fin de partie.

POUR JOUER :
On lance le programme par RUN. Un texte de présentation est affiché, puis le message "RESTE DANS LA DIVISION PAR 3" s'affiche.
On doit répondre 0, 1 ou 2, et l'on obtient le message "RESTE DANS LA DIVISION PAR 5" etc.
La solution est affichée dès que l'on a donné les 3 restes.
Pour une autre partie, taper O à la question VOULEZ-VOUS UN AUTRE ESSAI (O/N).
Tapez N pour finir.

EXTENSIONS POSSIBLES :
On peut étendre le programme aux entiers de 1 à 1000, en utilisant les restes par 3, 5, 7 et 11 ; on pourra rajouter du son et de la couleur pour rendre le programme plus attrayant.

```
 10 'DECOMAQUE
 20 CLS:SPECIAL:SCALE 1,1
 30 PRINT"VOUS DEVEZ ENTRER LES RESTES"
 40 PRINT"DANS LES DIVISIONS PAR 3,5,7"
 50 PRINT"D'UN NOMBRE ENTIER < 100"
 60 I=3:S=1
 70 CURSOR 0,4:BEEP
 80 PRINT"RESTE DANS LA DIV. PAR";I
 90 A$=INSTR$(1):IF A$<"0" OR A$>CHR$(47+I) THE
N BEEP:GOTO 90
 100 A(S)=ASC(A$)-48
 110 S=S+1:I=I+2
 120 IF I<8 THEN GOTO 70
 130 CLS:SCALE 1,1:PEN 3:PRINT"RESTE : PAR 3  P
AR 5  PAR 7"
 140 FOR I=1 TO 3
 150 PEN 1:CURSOR I*7+1,3:FOR J=1 TO I*2
 160 IF J<= A(I) THEN PRINT"*";:ELSE PRINT".";
 170 NEXT J:PRINT
 180 NEXT I
 190 T=A(1)*70+A(2)*21+A(3)*15
 200 IF T>105 THEN T=T-105:GOTO 200
 210 PEN 3:CURSOR 0,8:PRINT"LA SOLUTION EST :";
:PEN 1:PRINTT
 220 SCALE 1,1:CURSOR 0,22:PEN 2:PRINT"UN AUTRE
 ESSAI (O/N) :"
 230 A$=INSTR$(1):IF A$="O" THEN RUN
 240 STANDARD
```

Chasse au sous-marin

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexion.
Difficulté du jeu : moyenne.

LE JEU :
Vous êtes à bord d'un bâtiment de surface, chargé de traquer un sous-marin caché quelque part dans votre secteur ; vous devez vous déplacer jusqu'à venir au-dessus du sous-marin et tirer à une certaine profondeur ; mais, attention : le sous-marin, lui aussi, est mobile ; il ne coupera ses moteurs que lorsque vous serez exactement à sa verticale, pour ne pas être repéré.

LE PROGRAMME :
La boucle I (6Ø-12Ø) affiche le terrain.
La ligne 14Ø définit la position initiale du sous-marin ; la ligne 15Ø permet son déplacement, contrôlé par la ligne 16Ø (on reste dans les limites du terrain).
La ligne 18Ø affiche votre position.
Les lignes 22Ø à 24Ø contrôlent votre déplacement.
La distance est calculée en ligne 25Ø, affichée en ligne 26Ø et testée en lignes 27Ø et 28Ø.
Le test de fin de partie est effectué à la ligne 31Ø.

POUR JOUER :
On lance le programme par RUN et on entre la dimension ; la grille de jeu s'affiche. Vous êtes au départ en position (Ø, Ø) et vous vous déplacez avec les flèches.
Après chaque déplacement, la nouvelle distance du sous-marin s'affiche ; à chaque coup le sous-marin se déplace. Si vous réussissez à vous placer à sa verticale, la distance Ø est affichée et il ne bouge plus.
Vous tirez en donnant une profondeur entière (entre 2 et 4, bornes comprises). Si vous le ratez, il se replace n'importe où, et on recommence ; si vous le coulez, vous obtenez l'affichage BRAVO et la partie s'arrête.
On peut arrêter à tout moment le programme en utilisant les touches CNT et C.

EXTENSIONS POSSIBLES :
Il est possible de prévoir une amélioration du graphisme, d'améliorer les possibilités de déplacement du chasseur (un peu lent, dans cette version), de prévoir des parcours plus sophistiqués pour le sous-marin, de mettre un compteur de temps...

```
 10 'CHASSE AU SOUS-MARIN
 20 SEEDTIME(0)
 30 CLS
 40 INPUT"DIMENSION DE LA GRILLE :";H:IF H<1 OR
 H>5 THEN BEEP:GOTO 30
 50 CLS:SPECIAL:SCALE 1,1
 60 FOR I=0 TO H
 70 PEN 3:CURSOR 2*I+3,0:PRINTI
 80 CURSOR 0,2*I+2:PRINTI
 90 PEN 2:FOR J=0 TO H
 100 CURSOR 2*J+4,2*I+2:PRINT"."
 110 NEXT J
 120 NEXT I
 130 I=1:X=0:Y=0
 140 V=INT(RND(0,H)):U=INT(RND(0,H))
 150 U=U+I:V=V+I
 160 IF (U<0 OR U>H OR V<0 OR V>H) THEN I=-I
 170 CURSOR 2*Y+2,2*X+2: IF Y<>0 THEN PRINT"."
 180 CURSOR 2*Y+4,2*X+2:PEN 1:PRINT"#":PEN 2
 190 CURSOR 2*Y+6,2*X+2:IF Y<>H THEN PRINT"."
 200 CURSOR 2*Y+4,2*X:IF X<>0 THEN PRINT"."
 210 CURSOR 2*Y+4,2*X+4:IF X<>H THEN PRINT"."
 220 A$=INSTR$(1)
 230 Y=Y+(A$=CHR$(9))*(Y<H)-(A$=CHR$(8))*(Y>0)
 240 X=X+(A$=CHR$(30))*(X<H)-(A$=CHR$(29))*(X>0
)
 250 D=INT(SQR(ABS(X-V)^2+ABS(Y-U)^2))
 260 CURSOR 0,25:PRINT"DISTANCE ... :";D;"     "
;
 270 IF D=0 THEN GOTO 300
 280 IF D<3 THEN GOTO 150
 290 GOTO 170
 300 CURSOR 18,24:INPUT"PROF :";D
 310 IF D=INT(RND(2,5)) THEN LASER:PRINT:PRINT"
BRAVO .... ":STANDARD:END
 320 SHOTO:CURSOR 18,24:PRINT"         "
 330 GOTO 140
```

Clavier

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexe (apprentissage du clavier).
Difficulté du jeu : difficile.

LE JEU :
Une reproduction du clavier de votre ordinateur apparaît sur l'écran ; un point apparaît en dessous d'un caractère, qu'il convient de frapper au clavier. Plus l'on frappe vite, et plus le point se déplace rapidement (le temps écoulé depuis le début s'affiche en fin de partie).

LE PROGRAMME :
Une variable A$ dimensionnée en ligne 6Ø, contiendra la liste des touches concernées. Les valeurs des touches sont entrées de la ligne 7Ø à la ligne 12Ø par appel d'un sous-programme situé en 3ØØ.
Les boucles emboîtées J (13Ø-17Ø) et I (14Ø-16Ø) dessinent le clavier.
La ligne 18Ø définit la position du point imprimé par la ligne 19Ø.
La touche frappée est testée aux lignes 22Ø-23Ø et le point est effacé par la ligne 25Ø.
Les affichages de fin sont réalisés aux lignes 27Ø à 29Ø.

POUR JOUER :
On lance le programme par RUN. L'on dispose alors de quelques secondes pour frapper la touche qui est visualisée, puis une autre touche, etc.
La durée de la partie n'est pas limitée ; il suffit de taper Ø pour obtenir l'affichage du score, et du temps total.
Faire RUN pour une autre partie.

EXTENSIONS POSSIBLES :
On pourra envisager de faire clignoter les lettres. On pourra afficher les lettres devinées justes en bas de l'écran, et éventuellement s'entraîner à taper un texte qui défile en haut de l'écran, lettre par lettre.

```
 10 'CLAVIER
 20 SEEDTIME(0)
 30 CLS
 40 SPECIAL:SCALE 1,2:PEN 2
 50 T=0:S=0:P=0
 60 DIM A$(12,3):A$(0,1)="11":A$(0,2)="12":A$(0
,3)="10"
 70 S$="AZERTYUIOP=":J=1
 80 GOSUB 300
 90 S$="QSDFGHJKLM*/":J=2
 100 GOSUB 300
 110 S$="WXCVBN,;.?":J=3
 120 GOSUB 300
 130 FOR J=1 TO 3
 140 FOR I=1 TO VAL(A$(0,J))
 150 CURSOR2*I+J-2,J*2:PRINTA$(I,J)
 160 NEXT I
 170 NEXT J
 180 Y=INT(RND(1,4)):X=INT(RND(1,VAL(A$(0,Y))+1
))
 190 CURSOR 2*X+Y-2,Y*2:PAPER 3:PEN 1:PRINTA$(X
,Y)
 200 A$=INKEY$(150)
 210 BEEP
 220 IF A$=A$(X,Y) THEN S=S+1
 230 IF A$="0" THEN GOTO 270
 240 P=P+1
 250 CURSOR 2*X+Y-2,Y*2:PAPER 0:PEN 2:PRINTA$(X
,Y)
 260 GOTO 180
 270 SCALE 1,1:CURSOR 0,22:PRINTS;"JUSTES SUR";
P;"ESSAIS."
 280 PEN2:PAPER0:STANDARD
 290 END
 300 FOR I=1 TO VAL(A$(0,J)):A$(I,J)=MID$(S$,I,
1):NEXT I:RETURN
```

Le lézard d'Alice

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez difficile.

LE JEU :
Il s'inspire très vaguement d'une idée de LEWIS CARROLL, dans *Alice au Pays des Merveilles*. Dans l'histoire originale, Alice grandissait ou diminuait, alors que dans le jeu, pour des raisons d'encombrement de la mémoire, c'est un lézard qui change de taille.
Deux verres apparaissent sur l'écran : l'un contient le liquide qui fait grandir, l'autre le liquide qui fait rapetisser (malheureusement ce n'est pas toujours le même verre). Le joueur devra indiquer la quantité de liquide qu'il désire boire et le lézard changera de taille en fonction de cette quantité. Le but du jeu est de parvenir à lui rendre sa taille initiale.

LE PROGRAMME :
La boucle I (6Ø-9Ø) dessine les verres.
Les deux boucles emboîtées I (18Ø-22Ø) et J (19Ø-21Ø) les remplissent avec la quantité indiquée en ligne 17Ø.
La variation de taille du lézard est définie par la ligne 26Ø, avec contrôle par les lignes 27Ø et 28Ø.
La boucle I (29Ø-31Ø) dessine le corps du lézard, les lignes 32Ø à 37Ø dessinent l'œil et les pattes.
Un message gagnant sera affiché par la ligne 38Ø si l'on a trouvé la bonne taille.

POUR JOUER :
On lance le programme par RUN : les deux verres se dessinent, puis apparaît la question QUEL COTE à laquelle on répond en tapant G ou D ; on obtient l'affichage QUELLE QUANTITE ; on entre alors une quantité (de Ø à 13Ø) et on frappe ENTREE.
Le verre désigné se remplit sous vos yeux, puis, après quelques instants, le lézard est affiché. On devra alors frapper une touche quelconque ; si le lézard n'avait pas la bonne taille, aucun message n'est affiché et l'on revient au dessin des verres.

EXTENSIONS POSSIBLES :
Puisqu'il s'agit d'un jeu essentiellement graphique, il sera intéressant d'améliorer les dessins, d'introduire la couleur, de modifier le caractère aléatoire du choix des verres, etc.
Pourquoi ne pas relire *Alice au Pays des Merveilles*, les idées n'y manquent pas.

```
 10 'LE LEZARD D'ALICE
 20 SEEDTIME(0)
 30 N=18
 40 STANDARD
 50 WIPE:LITTLE
 60 FOR I=1 TO 9
 70 CURSOR 27,226-I*10:PRINTCHR$(127):CURSOR 69
,226-I*10:PRINTCHR$(127)
 80 CURSOR 87,226-I*10:PRINTCHR$(127):CURSOR 12
9,226-I*10:PRINTCHR$(127)
 90 NEXT I
 100 FOR I=3 TO 10:CURSOR 6*I+9,126:PRINTCHR$(1
27):NEXT I
 110 FOR I=13 TO 20:CURSOR 6*I+9,126:PRINTCHR$(
127):NEXT I
 120 CURSOR 33,100:PRINT"GAUCHE":CURSOR 93,100:
PRINT"DROIT"
 130 CURSOR140,196:PRINT"COTE (D/G) :";:
 140 A$=INSTR$(1):IF A$<>"D" AND A$<>"G" THEN B
EEP:GOTO 140
 150 U=2-20*(A$="D")
 160 PRINTA$
 170 CURSOR140,166:INPUT"QUANTITE    :";Q
 180 SOUND3,62:FOR I=1 TO 4*SQR(Q)
 190 FOR J=0 TO 35
 200 INK 1:PLOT 3*U+J+27,126+I
 210 NEXT J
 220 NEXT I
 230 HUSH:PAUSE.5
 250 H=2*(RND(0,1)<.5)+1
 260 N=N+H*INT(SQR(Q))
 270 IF N<12 THEN N=12
 280 IF N>58 THEN N=58
 290 FOR I=0 TO 4*N:SOUND 3,84
 300 INK3:FOR J=47TO50:PLOTI,J:NEXT
 310 NEXT I:HUSH
 320 T=4*(N-INT((SQR(N)/2)))
 330 INK1:PLOT T,59:PLOT T-1,58:PLOT T-2,57:PLO
T T+1,58:PLOT T+2,57:INK 2:PLOT T,56
 340 INK 1:T=4*SQR(N):PLOT T,45:PLOT T-1,44:PLO
T T-2,43
 350 T=T+2:PLOT T,45:PLOT T+1,44:PLOT T+2,43
 360 T=4*(N-SQR(N)):PLOT T,45:PLOT T-1,44:PLOT
T-2,43
 370 T=T+2:PLOT T,45:PLOT T+1,44:PLOT T+2,43
 380 IF N=28 THEN PRINT"BRAVO ...":STANDARD:END

 390 CURSOR 140,136:PRINT"TAPEZ UNE TOUCHE"
 400 A$=INSTR$(1):GOTO 50
```

Calculatrice

REFERENCES :
Difficulté de programmation : très difficile.
Catégorie de jeu : simulation d'une calculatrice.
Difficulté du jeu : moyenne.

LE JEU :
Il s'agit de la simulation, en affichage grand format, du fonctionnement d'une petite calculatrice de poche, de type 4 opérations. Les applications pédagogiques de ce programme sont évidentes et son fonctionnement très simple à maîtriser.

LE PROGRAMME :
Les variables B$ et C$, des lignes 2Ø et 3Ø, contiennent les informations codées qui permettront le dessin des chiffres en grand format.
Les lignes 6Ø à 15Ø s'occupent de la gestion des entrées.
La boucle I (16Ø-19Ø) réalise la représentation, en grand format, des chiffres sur l'écran, au fur et à mesure des entrées.
Le point décimal et le signe – sont gérés par les lignes 2ØØ et 21Ø.
Le nombre total des symboles entrés est contrôlé par la ligne 22Ø.
Le résultat est inscrit grâce à un programme fonctionnant en automatique qui se trouve aux lignes 27Ø à 39Ø (l'automatisme est géré par la ligne 23Ø, dans le programme principal).
Les sous-programmes placés aux lignes 4ØØ-41Ø, 42Ø-43Ø, 44Ø-45Ø et 46Ø-47Ø effectuent l'opération indiquée, selon la valeur de Z (ligne 31Ø).
Chaque résultat est mémorisé en A (F-1) (ligne 33Ø).

POUR JOUER :
On devra lancer le programme par RUN.
Puis, on tape les chiffres du premier nombre à entrer qui vont s'afficher un par un, au fur et à mesure (on ne pourra entrer plus de 8 symboles, y compris le signe – et le point décimal). La frappe d'un signe opératoire laisse l'affichage intact, mais l'introduction du prochain chiffre ou symbole provoque l'effacement immédiat et l'affichage du nouveau symbole.
Il est à remarquer que les chiffres s'introduisent de gauche à droite, sens de l'écriture usuelle.
L'appareil accepte, comme signes opératoires les symboles +, –, *, / ; le signe = provoquera l'affichage du résultat. Les opérations sont enchaînées automatiquement, le résultat précédent étant pris en compte comme opérateur du calcul suivant.
Pour faire une autre opération, il faudra taper CNT et C, puis RUN.
Taper CNT et C pour sortir du programme.

EXTENSIONS POSSIBLES :
On devra, tout d'abord, améliorer l'affichage du signe – (représenté, ici, par un trait), puis on pourra améliorer le dessin des chiffres, on peut aussi envisager une extension à plus de 8 chiffres, autoriser des opérations plus complexes (COS, SIN, ...), ajouter des mémoires, etc.

```
 10 'CALCULATRICE
 20 C$=CHR$(127):C$=C$+C$+C$+C$+"    "+C$+C$+"
"+C$
 30 B$="14441333331312113131441331213112141133331414114131"
 40 F=1:L=0:CLS:STANDARD:PEN2
 50 N=0:T=0:K=0:R$=""
 60 A$=INSTR$(1)
 70 IF K=1 THEN N=0:R$="":K=0
 80 IF A$="=" THEN GOTO 260
 90 X=(1 AND A$="+")+(2 AND A$="-")+(3 AND A$="
*")+(4 AND A$="/")
 100 R$=R$+A$
 110 IF X<>0 THEN GOTO 260
 120 IF R$="-" THEN A$=R$:GOTO 200
 130 IF A$<"." OR A$>"9" OR A$="/" THEN GOTO 60

 140 IF N=0 THEN CLS
 150 IF A$="." OR A$="-" THEN GOTO 200
 160 FOR I=1 TO 5
 170 S=3*VAL(MID$(B$,I+5*VAL(A$),1))-2
 180 CURSOR N*6+9,196-10*I:PRINTMID$(C$,S,3)
 190 NEXT I
 200 IF A$="-" THEN CLS:CURSOR 9,166:PRINTCHR$(
127)+CHR$(127)
 210 IF A$="." THEN CURSOR N*6+15,146:PRINTCHR$
(127)
 220 N=N-4*(N<28)
 230 IF T<>0 THEN T=T+1:IF T<=LEN(R$) THEN GOTO
 380
 240 IF K=1 AND L<>1 THEN GOTO 50
 250 T=0:GOTO 60
 260 IF R$="-" AND A(F)=0 THEN GOTO 120
 270 N=0:K=1
 280 IF (A$="*" OR A$="/") AND VAL(R$)=0 THEN A
(F)=1:GOTO 300
 290 A(F)=VAL(R$)
 300 IF X<>0 THEN Z=X
 310 ON Z GOSUB 400,420,440,460
 320 IF A$<>"=" THEN F=F+1:ELSE F=2
 330 A(F-1)=VAL(R$)
 340 IF VAL(R$)=0 THEN A(F-1)=A(F)
 350 R$=STR$(VAL(R$))
 360 IF LEFT$(R$,1)=" " THEN R$=MID$(R$,2)
 370 T=1
 380 A$=MID$(R$,T,1)
 390 GOTO 140
 400 R$=STR$(A(F-1)+A(F)):A(F)=VAL(R$):L=1
 410 RETURN
 420 IF F<>1 THEN R$=STR$(A(F-1)-A(F)):A(F)=VAL
(R$):L=1
 430 RETURN
 440 IF F<>1 THEN R$=STR$(A(F-1)*A(F)):A(F)=VAL
(R$):L=1
 450 RETURN
 460 IF F<>1 THEN R$=STR$(A(F-1)/A(F)):A(F)=VAL
(R$):L=1
 470 RETURN
```

Niveau 5

Nom du programme	*Thèmes et particularités*
Compose	Faire de la musique, c'est d'abord l'écrire...
Losanges	Pour vous aider à vous endormir...
Mathieu	Vous ne savez pas dessiner ? Laissez faire votre ordinateur.
Ardoise magique	Mettez au point vos créations graphiques, sans papier ni crayon.
Dessin de dés	Améliorez vos jeux de dés en incorporant ce programme.
Smokey	Fumez sans risque, avec ce programme original pour un joueur.
Course de voitures	Un jeu de réflexe pas trop difficile, pour un joueur.
Echecs	Un programme assez simplifié, pour deux joueurs.
Danger spatial	Serez-vous assez adroit pour éviter toutes les météorites ?
Tic Tac Toe	Un classique, mais pas sans intérêt (deux joueurs).
Horloge 2	Horloge 1, avec remise à l'heure.
Dames	Pour deux joueurs : classique.
Quatre à quatre	Un jeu de réflexion un peu difficile, pour deux joueurs.
Anagrammes	Un bon jeu de réflexe : pour deux joueurs.
Oeufs	Ne laissez pas faire la pondeuse diabolique (un joueur).
Taquin de lettres	Un proverbe est caché dedans : à vous de le découvrir.
Cibles	Un peu d'adresse, beaucoup de chance : très visuel.
Gouttes	Un véritable gag visuel. A exploiter.
Obstacles	Beaucoup d'obstacles à éviter dans ce jeu d'adresse pour un joueur.
Captures dans l'espace	Votre mission serait facile, sans ces damnées météorites.
Anomalie	Exercez votre perspicacité dans la reconnaissance des symboles.

Plusieurs en un	Un terrain de jeu et les pions, pour deux joueurs.
Parcours du cavalier	Très spectaculaire, et instructif à plus d'un titre.
Mur de briques	Un grand classique dans une version très complète.
Othello	Pour deux joueurs ; il retourne les pions à votre place.

Les techniques spécifiques de HECTOR

LECTURE DE LA COULEUR D'UN POINT DE L'ECRAN PAR POINT (X, Y).

HECTOR dispose d'une fonction particulière, facile à manipuler, qui donne la couleur d'un point de l'écran, on écrit POINT (X,Y) pour avoir la couleur du point se trouvant en X (coordonnée horizontale) et en Y (coordonnée verticale).

Le nombre renvoyé va de Ø à 3.

INSTRUCTIONS GRAPHIQUES .

INK permet de choisir la couleur du dessin.

PLOT X,Y affiche le point de coordonnées X,Y dans la couleur prédéfinie par INK.

BOX X1,Y1,X2,Y2 trace un rectangle plein délimité par les deux points (X1,Y1) et (X2,Y2).

INSTRUCTIONS SONORES .

SOUND X,Y génère un son qui ne peut être arrêté que par l'instruction HUSH.

TONE H,D génère un son de hauteur H et de durée D.

Compose

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : composition musicale.
Difficulté du jeu : aucune, a priori.

LE JEU :
Il s'agit, en fait, d'un programme capable de composer, de façon très élémentaire, de la musique, et ceci d'une façon complètement aléatoire. L'ordinateur va, dans un premier temps, dessiner une portée, puis il affiche une succession de notes (allant du DO grave au SI aigu) et donne leur nom. Une nouvelle série de notes sera ensuite affichée, et ainsi de suite.

LE PROGRAMME :
La boucle I (5Ø-7Ø) dessine la portée. La variable N$, en ligne 8Ø, contient les noms des notes.
La deuxième boucle I (9Ø-17Ø) permet de faire le choix d'une note de façon aléatoire, à la ligne 1ØØ ; son nom est calculé en ligne 11Ø et la note est dessinée par la ligne 12Ø.
La ligne 18Ø permet de recommencer en tapant ENTREE ou d'arrêter en tapant F. Le sous-programme 24Ø-27Ø met en tableau les fréquences des notes.

POUR JOUER :
Lancer l'exécution du programme en tapant RUN.
La portée sera alors rapidement affichée, puis les notes sont représentées, avec leur nom affiché en bas de l'écran.
Une pression sur la touche ENTREE provoquera l'affichage d'une nouvelle série de notes, et une pression sur la touche F provoque la fin de la partie.

EXTENSIONS POSSIBLES :
On pourra affiner les règles de composition, afficher davantage de notes, introduire des dièses et des bémols, améliorer le graphisme.

```
 10 'COMPOSE
 20 SEEDTIME(0)
 25 GOSUB240
 30 STANDARD:PEN2
 40 CLS
 50 FOR I=1 TO 5
 60 CURSOR 0,196-20*I:PRINT"-------------------
-------------------"
 70 NEXT I
 80 N$="SI LA SOLFA MI RE DO "
 90 FOR I=1 TO 12
 100 X=INT(RND(1,15))
 110 D$=MID$(N$,3*(X+7*(X>7))-2,3)
 120 PEN 1:C=3*I-2:L=X-1:GOSUB1000:PRINT"!":L=X
:GOSUB1000:PRINT"!":C=3*I-3:L=X+1:GOSUB 1000:PR
INT"O":PEN2
 140 C=3*I-3:L=18:GOSUB1000:PRINTD$
 160 TONE T(X),INT(12000/T(X))
 170 NEXT I
 180 CURSOR 0,32:PRINT"TAPER 'ENTREE' POUR CONT
INUER ":INPUT"              'F' POUR TERMINER ";R$

 190 IF R$<>"F" THEN GOTO 40
 200 END
 240 DIM T(14)
 250 DATA35,43,51,59,63,71,80,85,97,110,124,131
,148,168
 260 FOR I=1 TO 14:READ T(I):NEXT
 270 RETURN
 1000 CURSOR C*6+9,226-L*10
 1010 RETURN
```

Losanges

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : interlude.
Difficulté du jeu : aucune.

LE JEU :
Le programme va progressivement construire sur l'écran un losange, puis il va l'effacer progressivement, le reconstruire, et ainsi de suite.
Vous pourrez envisager de l'utiliser pour vous endormir, si les moutons ne vous suffisent plus.
Une version, très légèrement modifiée, vous est également proposée, qui vous permettra de faire apparaître des motifs aléatoires après la construction de chaque losange.

LE PROGRAMME :
Une première boucle K (5Ø-13Ø) définit les dimensions des tracés successifs.
La deuxième boucle I (6Ø-12Ø) trace des bandes de points, en utilisant des instructions PLOT (lignes 8Ø à 11Ø).
Si la variable S (ligne 7Ø) est à 1, alors il y aura un choix de couleurs aléatoire.

POUR JOUER :
On commence l'exécution du programme avec RUN, et on regarde.
On peut arrêter à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
On pourra introduire le son, imaginer sur des principes analogues divers programmes (kaléidoscope, tapisserie...).

```
10 'LOSANGES
20 SEEDTIME(0)
30 WIPE:LITTLE
40 S=0
50 FOR K=0 TO 40
60 FOR I=0 TO K
70 IF S<>0 THEN C=INT(RND(0,4)) :ELSE C=C+1:IF
C=4 THEN C=0
80 INK C:PLOT 128-K+I,128-I
90 PLOT 128-K+I,128+I
100 PLOT 128+K-I,128+I
110 PLOT 128+K-I,128-I
120 NEXT I
130 NEXT K
140 S=NOT S
150 GOTO 50
```

Mathieu

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : dessin aléatoire.
Difficulté du jeu : aucune.

LE JEU :
On utilise la définition d'une courbe en coordonnées polaires et la fonction PLOT pour obtenir sur l'écran des dessins aléatoires (de type fleur), plus ou moins élégants.
Les deux paramètres principaux A et B pourront être, si nécessaire, entrés à la main (instructions INPUT) pour faire une étude plus systématique des différentes courbes possibles.

LE PROGRAMME :
Les 3 premiers paramètres Z, K et T, introduits en ligne 4Ø, définissent le rayon maximum et l'occupation de l'écran.
Les variables aléatoires A et B, choisies entre Ø et 3, de la ligne 6Ø, définissent l'allure de la courbe ; il pourra être intéressant de les introduire manuellement (par exemple : 6Ø INPUT A, B).
La boucle H (7Ø-12Ø) réalise le tracé, point par point. Le rayon R est calculé en ligne 8Ø, et le passage en coordonnées cartésiennes est effectué aux lignes 9Ø et 1ØØ.
L'affichage des points est réalisé en ligne 11Ø.
La ligne 14Ø imprime un message de fin.
La ligne 16Ø permet de refaire un nouveau dessin, alors que la ligne 17Ø assure un nouveau tracé sans effacement du précédent.

POUR JOUER :
On lance le programme en tapant RUN.
Le dessin sera ensuite exécuté automatiquement.
Le message "c'est fini" signale la fin du tracé.
Si l'on veut un nouveau tracé, on tape N.
Si l'on veut un tracé sans effacement du dessin précédent, on tape A.
Pour arrêter, faire F.

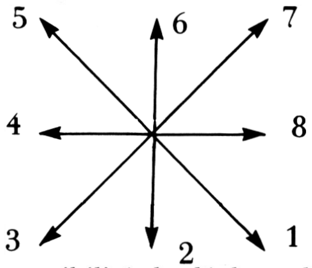
EXTENSIONS POSSIBLES :
On pourra essayer d'utiliser d'autres fonctions pour réaliser des tracés encore plus extraordinaires (par exemple avec simulation de la troisième dimension).

```
10 'MATHIEU
20 COVER 1:SEEDTIME(0)
30 WIPE:LITTLE:AC=0
40 Z=7:K=9:T=3
50 C=INT(RND(1,4)):IF C=AC THEN GOTO 50
60 A=RND(0,3):B=RND(0,3)
70 FOR H=0 TO 6.28 STEP 3.14/180
80 R=Z*SIN(H*T)
90 X=150+INT(K*R*COS(A*H))
100 Y=100+INT(K*R*SIN(B*H))
110 INK C:PLOT X,Y
120 NEXT H
130 HOME
140 PRINT"C'EST FINI !"
150 PRINT:INPUT"(A,N,F) :";R$
160 IF R$="N" THEN GOTO 30
170 IF R$="A" THEN AC=C:HOME:PRINT"
":PRINT:PRINT"                    ":GOTO 40
180 COVER 0
```

Ardoise magique

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : dessin.
Difficulté du jeu : facile.

LE JEU :
Comme le titre peut sembler l'indiquer, il s'agit, tout simplement d'utiliser l'écran du téléviseur comme une ardoise, en manipulant un point sur l'écran suivant 8 directions, à l'aide des touches de 1 à 8, suivant la convention :



On dispose aussi de la possibilité de déplacer le point sur l'écran sans laisser de trace en appuyant sur la touche Ø (ce qui permet l'effacement partiel) et de laisser de nouveau une trace en réappuyant sur Ø.
On peut choisir la couleur du point en appuyant sur une touche de A à D ; et il est possible d'effacer tout l'écran en appuyant sur P.

LE PROGRAMME :
La ligne 2Ø contient le choix des possibilités de couleur.
La ligne 5Ø imprime le point élémentaire.
Les lignes 6Ø à 9Ø contrôlent les entrées.
La ligne 13Ø appelle l'un des quatre sous-programmes effectuant le déplacement.
La ligne 14Ø vérifie que l'on ne sort pas des limites de l'écran.
La ligne 16Ø effectue l'effacement s'il y a lieu.
Le sous-programme 4ØØ-41Ø choisit la couleur ; 3ØØ-31Ø réalise ou non l'effacement.

POUR JOUER :
On lance le programme par RUN ; un point apparaît vers le milieu de l'écran. On devra taper Ø pour le déplacer sans afficher.
Si l'on frappe une lettre de A à D, la couleur du point change.
La touche P efface tout, et l'on recommence.
On peut sortir à tout moment, en tapant CNT et C.

EXTENSIONS POSSIBLES :
On pourra, bien sûr, envisager d'écrire un programme identique en utilisant des tracés de segments, prévoir un sous-programme qui permette la sauvegarde du dessin réalisé pour une utilisation ultérieure, etc.

```
10 'ARDOISE MAGIQUE
20 C$="0123"
30 WIPE:LITTLE
40 X=114:Y=114:C=2
50 INK C:PLOT X,Y
60 A$=INSTR$(1)
70 IF A$>="A" AND A$<="D" THEN GOSUB 400
80 IF A$="O" THEN GOSUB 300
90 IF A$="P" THEN GOTO 30
100 F=-2*(VAL(A$)<5)-1
110 A=X:B=Y
120 U=VAL(A$)+2*(F-1)
130 ON U GOSUB 1000,2000,3000,4000
140 D=-(X>0 AND X<254 AND Y>0 AND Y<227)
150 IF D=0 THEN GOSUB 5000
160 IF S<>0 THEN INK 0:PLOT A,B
170 GOTO50
300 S=ABS(S-1)
310 RETURN
400 C=VAL(MID$(C$,ASC(A$)-64,1))
410 RETURN
1000 Y=Y+F
2000 X=X+F
2010 RETURN
3000 GOSUB2000
4000 Y=Y-F
4010 RETURN
5000 X=A:Y=B
5010 RETURN
```

Dessin de dés

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

LE JEU :
Le programme proposé est surtout intéressant par son côté graphique puisqu'il permet de représenter, de façon très convenable, de 1 à 5 dés sur l'écran, en utilisant une technique de dessin un peu particulière.
Chaque réponse à la question NOMBRE DE DES provoque immédiatement un nouveau lancer du nombre de dés indiqué.

LE PROGRAMME :
La ligne 7Ø permet de contrôler s'il y a eu ou non action sur une touche de 1 à 5.
Les lignes 9Ø et 1ØØ initialisent les deux variables de dessin des dés.
La boucle I (11Ø-2ØØ) et les lignes 12Ø et 13Ø dessinent le cadre du dé.
La ligne 14Ø réalise le tirage, et la boucle J (15Ø-19Ø) dessine les points en relation avec le tirage réalisé (en appelant les variables A$ et B$).

POUR JOUER :
On lance le programme par l'ordre RUN, et, à la question NOMBRE DE DES, on répond en frappant une touche de 1 à 5. Les dés demandés sont alors représentés, et la question NOMBRE DE DES est de nouveau posée.
Il suffira de taper CNT et C pour sortir du programme.

EXTENSIONS POSSIBLES :
On pourra conserver la partie dessin utilisée dans ce programme pour réaliser toute sorte de jeux de dés, avec une représentation graphique de qualité convenable et facile à modifier si besoin était.

```
10 'DESSIN DE DES
20 SEEDTIME(0)
30 WIPE:LITTLE
40 CLS:COVER 1
50 CURSOR19,76:PRINT"NOMBRE DE DES : ";
60 A$=INSTR$(1):IF A$="0" THEN END
70 N=VAL(A$):IF N<1 OR N>5 THEN GOTO 40
80 PAPER 0
90 A$="          "+CHR$(127)+"   "+CHR$(127)+"
   "+CHR$(127)+CHR$(127)+"   "+CHR$(127)
100 B$="121314324515525555"
110 FOR I=1 TO N
120 BOX 54*(I-1)+15,216,54*(I-1)+57,146
140 X=INT(RND(1,7))
150 FOR J=2 TO 6
160 IF J=3 OR J=5 THEN CURSOR 54*I-33,226-J*10
:PRINT"     ":GOTO 190
170 S=5*VAL(MID$(B$,3*(X-1)+J/2,1))-4
180 CURSOR 54*I-33,226-J*10:PRINT MID$(A$,S,5)

190 NEXT J
200 NEXT I
210 GOTO50
```

Smokey

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

LE JEU :
Une colonne de fumée grise apparaît sur l'écran. Le joueur, représenté par une étoile rouge, devient visible sur la droite. Il doit dans un premier temps se positionner sur la colonne de fumée, puis à partir de là, il va automatiquement avaler la fumée, à condition qu'il réussisse à se maintenir sur sa trace.

LE PROGRAMME :
La ligne 9Ø fait apparaître la fumée au bas de l'écran ; la ligne 1ØØ permet de changer (vers la droite ou vers la gauche) la direction de la colonne, et la ligne 12Ø fait monter la colonne.
Le joueur apparaît grâce à la ligne 14Ø, et son déplacement est effectué par les lignes 18Ø et 19Ø.
Les lignes 25Ø et 26Ø se chargent de l'affichage du résultat.

POUR JOUER :
On doit lancer l'exécution du programme par RUN.
La fumée commence à s'élever, tandis que le joueur, apparu sous forme d'une étoile rouge dans un coin de l'écran, va se déplacer sur elle à l'aide des 4 flèches. Dès qu'il arrive sur la fumée, il va automatiquement commencer à l'avaler, et devra essayer de suivre la colonne sans dévier (grâce aux flèches). Il gagne s'il arrive en bas de l'écran avant que la fumée n'arrive en haut.
L'affichage final indique alors BRAVO ou PERDU, ainsi que le score obtenu.

EXTENSIONS POSSIBLES :
On pourra, tout en conservant l'idée de départ, imposer d'autres règles : par exemple avaler toute la fumée, qui pourra aussi repartir d'un autre endroit si l'on dévie de la colonne initiale.
On pourra aussi améliorer les contrôles de fin de partie, et la qualité du graphisme.

```
10 'SMOKEY
20 seedtime(0)
30 standard:paper0:pen2:cover 1
40 X$=chr$(127)
50 J$="*"
60 S=0:X=21:Y=int(rnd(0,24))
70 U=8:V=13
80 cls
90 beep:pen 2:cursor Y*6+9,16:printX$
100 Y=Y+(rnd(0,1)<.5)*(Y<24)-(rnd(0,1)<.5)*(Y>0)
110 T=1
120 X=X-T
130 if X=0 then goto 250
140 cursor V*6+9,226-U*10:pen 1:printJ$ ;
150 if T<>0 then A$=inkey$(60)
170 cursorV*6+9,226-U*10:print" ";
180 U=U-(A$=chr$(30))*(T=1)*(X<24)
190 U=U-(T=0)
200 V=V+(A$=chr$(9))*(V<24)-(A$=chr$(8))*(V>0)
210 C=point(V*6+11,212-U*10)
220 if C<>2 then goto 90
225 laser
230 T=0:S=S+1:A$=""
240 if U<21 then goto 140
250 if T=0 then print"BRAVO ...":goto 270
260 print"PERDU ..."
270 laser:print"SCORE :";S
280 cover 0
```

Course de voitures

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexe (1 joueur).
Difficulté du jeu : moyenne.

LE JEU :
Vous circulez sur une route très sinueuse, ne comportant aucun obstacle, et vous devez essayer de la suivre le plus longtemps possible. La trace de votre parcours reste affichée en permanence, et votre score est affiché dès que vous avez un accident.

LE PROGRAMME :
L'impression de la route est assurée par les lignes 8Ø à 1ØØ.
La ligne 11Ø fait avancer la route, et la ligne 12Ø imprime la voiture.
La ligne 15Ø contrôle son mouvement, et la ligne 16Ø détermine si la route fait ou non des virages.
La ligne 17Ø regarde si l'on a touché le bord de la route.
La ligne 18Ø incrémente le score, et la ligne 19Ø efface l'ancienne position de la voiture (la couleur du fond apparaît à l'emplacement de sa trajectoire).

POUR JOUER :
Lancer l'exécution du programme par l'ordre RUN, le dessin de la route commence à évoluer, et votre véhicule, représenté par une voiture, apparaît.
Pour le déplacer vers la droite ou vers la gauche, vous devrez utiliser les touches → et ← (flèches de déplacement du curseur). Il vous faudra éviter le plus longtemps possible de toucher une bordure noire, ce qui provoquera l'arrêt du programme et l'affichage de la mention ACCIDENT, suivie de l'indication sur le score réalisé.
On pourra arrêter le déroulement de la partie à tout moment en pressant les touches CNT et C.

EXTENSIONS POSSIBLES :
On peut envisager de jouer à plusieurs, chacun son tour, en enregistrant le meilleur score.
On peut aussi envisager d'élargir la piste, et de faire courir plusieurs voitures.
On pourra aussi s'inspirer des jeux d'arcades qui sont particulièrement nombreux sur ce thème.

```
 10 'COURSE DE VOITURES
 20 SEEDTIME(0)
 30 STANDARD:PAPER0:PEN2:COVER 1
 40 X$="#"
 50 T=0:CLS
 60 Y=11:A=5:B=10:S=0
 70 BEEP
 80 CURSORY*6+9,36:PEN2:PRINT CHR$(127)+CHR$(12
7);
 90 PEN1:PRINT CHR$(127)+CHR$(127)+CHR$(127);
 100 PEN2:PRINT CHR$(127)+CHR$(127)
 110 SCROLL7*64
 120 CURSOR B*6+9,226-A*10:PEN 3:PRINTX$
 130 A$=INKEY$(20)
 140 C=B
 150 B=B-(A$=CHR$(9))+(A$=CHR$(8))
 160 Y=Y+(2*RND(0,1)AND Y<25)-(2*RND(0,1) AND Y
>3)
 170 IF POINT(B*6+11,212-A*10)=0 AND S>=18 THEN
 GOTO 210
 175 IF POINT(B*6+11,212-A*10)=2 THEN BELL
 180 S=S+1
 190 CURSOR C*6+9,226-A*10:PRINT" "
 200 GOTO 70
 210 SHOT1:SHOT0:SIREN:PRINT"ACCIDENT ... SCORE
:";S
 220 COVER 0:PEN 2
 230 PAUSE 5:HUSH
```

Echecs

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexion, pour deux joueurs.
Difficulté du jeu : difficile.

LE JEU :
Le programme propose seulement à deux joueurs de disputer une partie en se servant de l'écran du téléviseur comme plateau de jeu. Il n'y a pas de contrôle de la validité des coups, mais seulement gestion de la position des pièces sur l'échiquier. Les pions et les pièces sont désignés par leur initiale. La couleur permet de distinguer chaque camp.

LE PROGRAMME :
Les lignes 60 à 200 dessinent l'échiquier. Les lignes 210 et 220 définissent les conditions initiales, à l'aide de deux tableaux : R (I, J) pour les codes des pièces, et C (I, J) pour la couleur.
Les boucles emboîtées J (240-290) et I (250-280) dessinent les pièces.
La ligne 320 permet l'entrée de la ligne et de la colonne de départ, avec contrôle en ligne 350.
La ligne 370 permet d'entrer colonne et ligne d'arrivée.
La ligne 400 note la nouvelle position.
La ligne 410 assure le changement de joueur.

POUR JOUER :
On lance l'exécution par RUN.
L'échiquier est dessiné, puis les pièces, et enfin l'inscription JOUEUR BLEU suivie de DEPART (C et L) : on entre la colonne et la ligne de départ, sous la forme E 7 par exemple, on obtient alors l'affichage de ARRIVEE (C et L) et on répond de même. Le déplacement est alors effectué, et c'est au tour de l'autre joueur.
Pour arrêter, il suffit de taper CNT et C.

EXTENSIONS POSSIBLES :
On pourra chercher à réaliser un graphisme plus précis, introduire le contrôle du mouvement des pièces. Et pourquoi pas, écrire un programme qui permette à HECTOR de se comporter en partenaire honorable. Bon courage !

```
 10 'ECHECS
 20 CLS1:PEN0:COVER1
 30 A$=CHR$(127)+CHR$(127)+CHR$(127)
 40 J$(0)="F":J$(1)="R":J$(2)="T":J$(3)="D":J$(
4)="P"
 50 J$(5)="C":J$(6)="
 60 PRINT"      A  B  C  D  E  F  G  H":PRINT
 70 FOR I=1 TO 8
 80 FOR J=1 TO 2
 90 IF J=2 THEN PRINTI;" ";:ELSE PRINT"    ";
 100 PAPER2
 110 FOR K=1 TO 8
 120 IF (I+K)/2<>INT((I+K)/2) THEN PRINTA$;:ELS
E PRINT"   ";
 130 IF I=2 OR I=7 THEN R(I,K)=4
 140 IF I<>2 AND I<>7 THEN R(I,K)=6
 150 C(I,K)=0
 160 IF I=7 OR I=8 THEN C(I,K)=2
 170 NEXT K
 180 PAPER1:PRINT
 190 NEXT J
 200 NEXT I
 210 R(1,1)=2:R(1,2)=5:R(1,3)=0:R(1,4)=3:R(1,5)
=1:R(1,6)=0:R(1,7)=5:R(1,8)=2
 220 FOR I=1 TO 8:R(8,I)=R(1,I):NEXT I
 230 H=1
 240 FOR J=1 TO 8
 250 FOR I=1 TO 8
 260 X=3*J+2:Y=2*I+1:F=POINT(X*6+6,231-Y*10)
 270 CURSOR X*6+9,226-Y*10:PEN C(I,J):PAPER 2-C
(I,J):IF R(I,J)=6 THEN PAPER F
 275 PRINTJ$(R(I,J))
 280 NEXT I
 290 NEXT J
 300 PAPER1:PEN0:CURSOR180,200:PRINT"        ":CU
RSOR 180,180:PRINT"        ":CURSOR 180,160:PRIN
T"        "
 310 CURSOR 180,200:IF H=1 THEN PRINT"BLANC":EL
SE PRINT"NOIR"
 320 CURSOR 180,180:INPUT"DEP.";D$
 330 '
 340 GOSUB 500
 350 IF R(L,C)=6 OR (C(L,C)=2 AND H=0) OR (C(L,
C)=0 AND H=1) THEN BEEP:GOTO 310
 360 T=R(L,C):S=C(L,C):R(L,C)=6
 370 CURSOR 180,160:INPUT"ARR.";D$
 380 '
 390 GOSUB 500
 400 R(L,C)=T:C(L,C)=S
 410 H=ABS(H-1)
 420 GOTO 240
 500 L=VAL(RIGHT$(D$,1)):C=ASC(LEFT$(D$,1))-64
 510 RETURN
```

Danger spatial

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexe, pour un joueur.
Difficulté du jeu : moyenne.

LE JEU :
Vous êtes aux commandes d'un engin spatial qui vient de pénétrer dans un nuage de météorites. Vous devez essayer de franchir le nuage sans heurter une seule météorite. Votre score sera affiché en fin de partie, lorsque vous entrez en collision avec une météorite.

LE PROGRAMME :
La position de chaque météorite est déterminée par les lignes 6Ø et 8Ø ; elle est imprimée par les lignes 7Ø et 9Ø.
L'avance des météorites est réalisée par la ligne 1ØØ.
Les lignes 1ØØ et 11Ø vérifient si vous avez ou non heurté une météorite.
La ligne 13Ø dessine votre vaisseau spatial, et la ligne 16Ø permet de le déplacer.
Le compteur se trouve en ligne 17Ø.

POUR JOUER :
Le jeu est lancé par l'ordre RUN.
Votre vaisseau apparaît vers le milieu de l'écran, et les météorites viennent à votre rencontre en partant du bas de l'écran. Pour vous déplacer, vous pouvez utiliser les touches de déplacement horizontal (→ et ←).
On peut arrêter la partie à tout moment en tapant CNT et C. Sinon, le jeu s'arrête lorsque vous heurtez une météorite, et le score est alors affiché.

EXTENSIONS POSSIBLES :
Il serait intéressant de rendre le jeu de plus en plus difficile en faisant apparaître de plus en plus de météorites au fur et à mesure du déroulement de la partie.
On peut aussi essayer de détruire les météorites en les pulvérisant au fur et à mesure de leur apparition.

```
 10 'DANGER SPATIAL
 20 SEEDTIME(0)
 30 STANDARD:PAPER0:PEN2:COVER 1
 40 CLS
 50 X=18:Z=0:Y=X
 60 H=INT(RND(9,232))
 70 IF RND(0,1)<.7 THEN BEEP:CURSOR H,36:PEN 1:
PRINTCHR$(127)
 80 K=INT(RND(9,232))
 90 IF RND(0,1)<.6 THEN BEEP:CURSOR K,36:PEN 1:
PRINTCHR$(127)
 100 SCROLL10*64:FOR I=0TO11:B=POINT(X*6+9+I,92
):IF B<>0 THEN I=11
 105 NEXT I
 110 IF B=1 OR C=1 THEN GOTO 190
 120 CURSOR Y*6+3,102:PRINT"     ":CURSOR Y*6+3,
112:PRINT"     "
 130 CURSOR X*6+9,92:PEN 3:PRINT"##"
 140 A$=INKEY$(30)
 150 Y=X
 160 X=X+(A$=CHR$(9))*(X<36)-(A$=CHR$(8))*(X>0)

 170 Z=Z+1
 180 GOTO 60
 190 LASER:LASER:LASER
 200 PRINT:PRINT"LE SCORE EST DE";Z
 210 COVER 0
```

Tic tac toe

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (deux joueurs).
Difficulté du jeu : facile.

LE JEU :
Un grand classique des jeux d'écoliers, proposé ici dans une version ne comportant pas le jeu machine : chaque joueur devra placer à son tour un 0 ou un * dans une case libre de la grille de jeu, et le premier qui réussit un alignement de 3 symboles identiques a gagné (l'alignement peut être horizontal, vertical, ou suivant l'une des deux diagonales).

LE PROGRAMME :
Le terrain de jeu est dessiné à l'aide d'une boucle I (40-80).
La ligne 100 s'occupe d'afficher le symbole cœur ou carreau correspondant au joueur qui a le trait.
Un symbole clignotant (ligne 130) est affiché en haut et à gauche du terrain, même si la case est déjà occupée : les lignes 160 et 170 permettent de déplacer le symbole sur le terrain de jeu.
La ligne 180 vérifie si la case où l'on veut jouer est libre, et si l'on désire jouer.
La ligne 190 affiche le coup, et la ligne 200 réalise le changement de joueur.

POUR JOUER :
Le terrain de jeu sera affiché dès la frappe de RUN.
Un symbole clignotant apparaît alors en haut et à gauche de l'écran ; le joueur peut le déplacer sur le terrain de jeu à l'aide des 4 flèches (touches ←, → et ↑, ↓).
Lorsqu'il arrive sur la case où il désire jouer, il suffit qu'il appuie sur la touche 0, le coup sera refusé si la case n'est pas libre.
Si la case est libre, le symbole qui correspond à ce joueur est inscrit, et on change de joueur.
Lorsque la partie est finie, il convient de taper CNT et C pour sortir du programme. On pourra ensuite faire une autre partie avec RUN.

EXTENSIONS POSSIBLES :
On pourrait réaliser après chaque coup un contrôle pour vérifier si l'un des joueurs a gagné. On peut aussi réaliser un jeu machine mais il faudrait éviter qu'il soit par trop systématique (par exemple, la machine jouant toujours son premier coup sur la case centrale).

```
 10 'TIC TAC TOE
 20 WIPE:INK2:BOXO,228,112,17
 30 SPECIAL:SCALE 1,2:PAPERO:PEN2:J$(O)="*":J$(
1)="O"
 40 FOR I=1 TO 3
 50 FOR J=1 TO 3
 60 CURSOR 4*I-2,4*J-2:PAPERO:PRINT" ":
 70 NEXT J
 80 NEXT I
 90 K=O
 100 CURSOR14,2:PEN 1+2*K:PRINT"JOUEUR ";J$(K)
 110 I=1:J=1
 120 C=POINT(I*32-8,259-J*64)
 130 CURSOR4*I-2,4*J-2:PAPER 2:PRINT" "
 135 PAUSE.15
 140 CURSOR4*I-2,4*J-2:IF C=O THEN PAPERO:PRINT
" ":ELSE PAPERC:PEN O:PRINTJ$((C-1)/2)
 150 A$=INKEY$(40)
 160 J=J+(A$=CHR$(30))*(J<3)-(A$=CHR$(29))*(J>1
)
 170 I=I+(A$=CHR$(9))*(I<3)-(A$=CHR$(8))*(I>1)
 175 IF C<>O AND A$="O" THEN BELL
 180 IF C<>O OR A$<>"O" THEN GOTO 120
 190 CURSOR 4*I-2,4*J-2:PAPER K*2+1:PEN O:PRINT
J$(K)
 200 K=ABS(K-1)
 210 GOTO100
 220 GOTO 100
```

Horloge 2

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : simulation.
Difficulté du jeu : aucune.

LE JEU :
Ce programme simule une horloge digitale grand format en utilisant les caractères d'origine de HECTOR, notablement agrandis. Il présente la particularité de permettre la mise à l'heure, et éventuellement l'arrêt ou le départ de l'horloge.

LE PROGRAMME :
Jusqu'à la ligne 3ØØ, il est identique à HORLOGE 1. La ligne 36Ø affiche le mode d'emploi.
Les lignes 39Ø-49Ø permettent la mise à jour des heures, et les lignes 5ØØ-6ØØ des minutes.

POUR JOUER :
Le programme vous demande l'heure de départ, puis affiche l'horloge. En tapant ENTREE, vous pouvez la remettre à l'heure grâce aux flèches ← et →.
Faire CNT et C pour arrêter le programme. On pourra régler l'horloge en jouant sur l'instruction 26Ø.

EXTENSIONS POSSIBLES :
On peut afficher les secondes, ajouter une possibilité de réveil, prévoir un agenda automatique, faire apparaître un coucou chantant au moment du passage de l'heure.

```
 10 'HORLOGE 2
 20 CLS:SPECIAL:SCALE1,2
 30 INPUT"HEURE (FORME HHMM) :";A$
 40 CLS1:PAPER 3
 50 K=1
 60 A=VAL(MID$(A$,K,1))
 70 IF A<>1 AND A<>4 THEN CURSOR 6*K-5,1:PRINT"
   "
 80 FOR J=1 TO 5
 90 IF A=0 OR (A>3 AND A<7) THEN CURSOR 6*K-5,
J:PRINT" "
 100 IF A<>5 AND A<>6 THEN CURSOR 6*K-3,J:PRINT
" "
 110 IF A=0 OR A=2 OR A=6 OR A=8 THEN CURSOR 6*
K-5,J+4:PRINT" "
 120 IF A<>2 THEN CURSOR 6*K-3,J+4:PRINT" "
 130 NEXT J
 140 IF A>1 THEN CURSOR 6*K-4,5:PRINT"   "
 150 IF A<>1 AND A<>4 AND A<>7 THEN CURSOR 6*K-
5,9:PRINT"    "
 160 K=K+1
 170 IF K<5 THEN GOTO 60
 180 B$=A$:A=1+VAL(A$)
 190 IF A=2360 THEN A=0
 200 A$=STR$(A)
 210 A$=RIGHT$("000"+MID$(A$,2),4)
 220 IF RIGHT$(A$,2)="60"THENA=A+40:GOTO200
 230 H=1
 240 FOR J=1 TO 120
 260 CURSOR 11,9:PENO:IF H=1 THEN PRINT":":ELSE
 PRINT" "
 270 K$=INKEY$(63):IF ASC(K$)=13 THEN J=120:NEX
TJ:GOTO 350
 280 H=ABS(H-1):IF H=1 THEN BEEP
 290 NEXT J
 300 LASER:GOTO 40
 350 STANDARD:PEN2:PAPERO:COVER 1:CLS
 360 PRINT"VOUS POUVEZ REMETTRE A L'HEURE":PRIN
T:PRINT"SUCCESSIVEMENT HEURES PUIS MINUTES.":PR
INT:PRINT"UTILISEZ LES FLECHES <- ET ->":PRINT"
PUIS 'ENTREE' POUR VALIDER."
 370 SPECIAL:SCALE 1,2:Y$=LEFT$(B$,2):Z$=RIGHT$
(B$,2):Y=VAL(Y$):Z=VAL(Z$)
 380 PEN1:PAPERO:CURSOR 8,10:PRINT" ";Y$;"   ";Z
$;" "
 390 PENO:PAPER1:CURSOR 9,10:PRINTY$
 400 X$=INSTR$(1)
 410 X=ASC(X$)
 420 IF X<>8 THEN GOTO 450
 430 Y=Y-1:IF Y=-1 THEN Y=23
 440 Y$=RIGHT$(STR$(100+Y),2):GOTO 390
 450 IF X<>9 THEN GOTO 480
 460 Y=Y+1:IF Y=24 THEN Y=0
 470 Y$=RIGHT$(STR$(100+Y),2):GOTO 390
 480 IF X<>13 THEN BEEP:GOTO 400
 490 PEN1:PAPERO:CURSOR 9,10:PRINTY$
 500 PENO:PAPER1:CURSOR 13,10:PRINTZ$
 510 X$=INSTR$(1)
 520 X=ASC(X$)
 530 IF X<>8 THEN GOTO 560
 540 Z=Z-1:IF Z=-1 THEN Z=59
 550 Z$=RIGHT$(STR$(100+Z),2):GOTO 500
 560 IF X<>9 THEN GOTO 590
 570 Z=Z+1:IF Z=60 THEN Z=0
 580 Z$=RIGHT$(STR$(100+Z),2):GOTO 500
 590 IF X<>13 THEN BEEP:GOTO 510
 600 A$=Y$+Z$
 610 GOTO 40
```

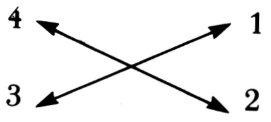
Dames

REFERENCES :
Difficulté de programmation : très difficile.
Catégorie de jeu : deux joueurs.
Difficulté du jeu : difficile.

LE JEU :
C'est le jeu classique de dames à la française, mais un peu incomplet puisqu'il ne prévoit pas la possibilité de faire des dames. La règle du jeu reste classique, mais les contrôles ont été un peu simplifiés. Les mouvements des pièces sont entrés en une seule manœuvre (position de départ et direction du déplacement) et les prises sont réalisées automatiquement, si elles sont possibles.

LE PROGRAMME :
Les boucles I (8Ø-17Ø) et J (9Ø-16Ø) dessinent le damier et définissent les positions initiales des pions noirs et des pions blancs. Les positions sont dans une table R (X, Y) et la couleur dans une autre table C (X, Y).
L'impression des pions proprement dite est effectuée par les boucles emboîtées J (19Ø-25Ø) et I (2ØØ-24Ø).
La ligne 27Ø indique le joueur qui a le trait.
Le coup est entré en ligne 27Ø.
La ligne 28Ø réalise le changement de joueur (on tape Ø).
La ligne 3ØØ regarde si le déplacement est possible.
La ligne 33Ø traite le cas d'un déplacement et la ligne 34Ø celui d'un saut.
La ligne 38Ø assure le changement de joueur.

POUR JOUER :
Dès le lancement du programme par RUN, le damier est affiché avec ses pions, et un caractère * de couleur apparaît en dessous pour indiquer que le trait appartient au joueur correspondant.
On joue en indiquant par un mot de 3 lettres (pouvant commencer par un Ø) la ligne et la colonne de la pièce à déplacer, puis la direction du déplacement (codée de 1 à 4) suivant la convention :



Par exemple 9D1 signifie que le pion situé en ligne 9, colonne D, se déplace dans la direction 1 (ne pas oublier de faire ENTREE).
De même ØG3 signifiera pièce noire, ligne Ø, colonne G, direction 3.
Le programme ne contrôle pas l'arrivée à dame, et n'interdit pas les déplacements en arrière.

EXTENSIONS POSSIBLES :

Il sera assez facile de commencer par améliorer les contrôles (changement de joueur, coup en arrière qui ne correspondrait pas à une prise, sortie de terrain). On pourra envisager la création de dames, leur déplacement, et éventuellement écrire un programme permettant à HECTOR de se mêler avec de bonnes chances de réussite à vos passionnantes parties.

```
 10 'DAMES
 20 STANDARD:PEN2:PAPER0:CLS:COVER 1
 60 CLS
 70 PRINT"     A  B  C  D  E  F  G  H  I  J"
 80 FOR I=1 TO 10
 90 FOR J=1 TO 10
 100 IF (I+J)/2=INT((I+J)/2) THEN PEN 1:ELSE PE
N 3
 110 CURSOR J*18+9,236-I*20:PRI":PRINTI-1
:CURSOR J*18+9,226-I*20:PRI"
 120 R(I,J)=0
 130 IF (I+J)/2<>INT((I+J)/2) THEN GOTO 160
 140 IF J<5 THEN R(I,J)=1:C(I,J)=3
 150 IF J>6 THEN R(I,J)=1:C(I,J)=1
 160 NEXT J
 170 NEXT I
 180 H=1
 190 FOR J=1 TO 10
 200 FOR I=1 TO 10
 210 IF (I+J)/2<>INT((I+J)/2) THEN GOTO 240
 220 X=3*J+1:Y=2*I
 230 CURSOR X*6+9,226-Y*10:PEN C(I,J)-2*(C(I,J)
=0)-1:PAPER 1:PRINT"*"
 240 NEXT I
 250 NEXTJ
 260 CURSOR207,180:PRINT"      "
 270 BEEP:CURSOR 219,200:PAPER1:PEN H*2:PRINT"*
":PAPER0:PEN2:CURSOR 207,180:INPUT":";D$
 280 IF D$="0" THEN H=ABS(H-1):GOTO 260
 290 GOSUB 400
 300 IF R(L,C)=0 OR (C(L,C)=1 AND H=1) OR (C(L,
C)=3 AND H=0) THEN BEEP:GOTO 260
 310 IF D=1 OR D=2 THEN T=1:ELSE T=-1
 320 IF D=2 OR D=3 THEN S=1:ELSE S=-1
 330 IF R(L+S,C+T)=0 THEN R(L,C)=0:C(L,C)=0: R(
L+S,C+T)=1:C(L+S,C+T)=1-2*(H=1):GOTO 380
 340 IF C(L+S,C+T)=1-2*(H=0) AND R(L+S,C+T)=1 A
ND R(L+2*S,C+2*T)=0 THEN GOTO 360
 350 BEEP:GOTO 260
 360 R(L,C)=0:C(L,C)=0:R(L+S,C+T)=0:C(L+S,C+T)=
0:R(L+2*S,C+2*T)=1:C(L+2*S,C+2*T)=1-2*(H=1)
 370 GOTO 190
 380 H=ABS(H-1)
 390 GOTO 190
 400 L=VAL(LEFT$(D$,1))+1:C=ASC(MID$(D$,2,1))-6
4:D=VAL(RIGHT$(D$,1))
 410 RETURN
```

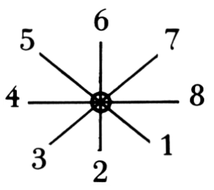
Quatre à quatre

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (un ou deux joueurs).
Difficulté du jeu : difficile.

LE JEU :
Il pourra être pratiqué à un ou plusieurs joueurs. Dans les deux cas, le but est d'arriver à échanger les pions X avec les pions O ; si l'on joue en solitaire, cet échange devra être réalisé avec le moins de coups possible. Les pions sont déplacés alternativement en respectant les règles de déplacement du cavalier aux échecs ; il n'y a pas de prise, et on ne peut passer son tour.

LE PROGRAMME :
La variable alphanumérique A$, en ligne 4Ø, contient la table des coups possibles (déplacement du cavalier).
La boucle I (6Ø-1ØØ) dessine le terrain de jeu, et les positions de départ.
La ligne 12Ø permet d'afficher O ou X, selon le joueur qui a le trait.
La ligne 15Ø permet l'affichage du curseur clignotant (ici un pavé de couleur) qui indique sur quelle case on se trouve.
Les lignes 18Ø et 19Ø permettent le déplacement du curseur.
La ligne 2ØØ regarde si l'on a appuyé sur une touche de 1 à 8, donnant la direction du déplacement.
La ligne 21Ø réimprime la figure qui était sous le curseur. La ligne 24Ø transforme en nombre le symbole chiffre frappé. La ligne 23Ø réalise le contrôle de départ.
Les lignes 25Ø et 26Ø effectuent le déplacement. La ligne 29Ø réalise le contrôle d'arrivée. Le résultat est imprimé par la ligne 3ØØ, le changement de joueur est réalisé par la ligne 31Ø et un "." est imprimé à l'emplacement de départ. Le sous-programme 33Ø-34Ø est utilisé pour les divers contrôles.

POUR JOUER :
On lance le programme par RUN. Le terrain de jeu est affiché, ainsi que le symbole permettant d'identifier le joueur qui a le trait. Le joueur peut choisir le pion qu'il désire déplacer en manœuvrant le curseur clignotant grâce aux touches de déplacement du curseur.
Pour déplacer son pion, il suffit, lorsque le curseur est sur ce pion, de taper la direction du déplacement, suivant le code :



Le coup sera refusé si la case de départ n'est pas conforme ou si la case d'arrivée est occupée. Sinon, le coup est réalisé, et le trait passe à l'autre joueur, qui déplace le curseur clignotant pour l'amener sur l'un des pions, et joue de la même façon.
Le joueur gagnant est celui qui réussit à amener le premier tous ses pions dans les cases initialement occupées par l'adversaire. On peut interrompre la partie à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
On peut améliorer les contrôles, prévoir un test de fin, utiliser un terrain de jeu différent ; on pourra aussi envisager l'écriture d'un programme permettant à HECTOR de participer au jeu avec de bonnes chances de l'emporter.

```
 10 'QUATRE A QUATRE
 20 CLS:SPECIAL:SCALE1,1
 40 H=1:A$="5665322356"
 50 CURSOR 4,2:PRINT"1 2 3 4 5"
 60 FOR I=1 TO 5
 70 PEN2:PAPER0:CURSOR 0,3*I+2:PRINTRIGHT$(STR$
(I),1):CURSOR 4,3*I+2:PRINT". . . . ."
 80 IF I<3 THEN PEN1:CURSOR 4,3*I+2:PRINT"* *"
 90 IF I>3 THEN PEN3:CURSOR 10,3*I+2:PRINT"O O"

 100 NEXT I
 110 X=1:Y=1
 120 PEN0:CURSOR 0,24:PAPER 1-2*(H=0):PRINT"JOU
EUR ";:IF H=0 THEN PRINT"O":ELSE PRINT"*"
 130 PAPER0:GOSUB330
 140 I=J
 150 CURSOR 2*X+2,3*Y+2:PRINT" "
 160 A=X:B=Y
 170 R$=INKEY$(80)
 180 Y=Y+(R$=CHR$(30))*(Y<5)-(R$=CHR$(29))*(Y>1
)
 190 X=X+(R$=CHR$(9))*(X<5)-(R$=CHR$(8))*(X>1)
 200 IF R$<"9" AND R$>"0" THEN GOTO 230
 210 TONE100,40:CURSOR 2*A+2,3*B+2:PEN J:IF J=1
 THEN PRINT"*":ELSE IF J=3 THEN PRINT"O":ELSE P
EN 2:PRINT"."
 220 GOTO 120
 230 IF (I=0) OR (I=1 AND H=0) OR (I=3 AND H=1)
 THEN BEEP:GOTO 210
 240 J=VAL(R$)
 250 S=Y:Y=Y+VAL(MID$(A$,J,1))-4
 260 T=X:X=X+VAL(MID$(A$,J+2,1))-4
 270 IF X<1 OR X>5 OR Y<1 OR Y>5 THEN Y=S:X=T:A
=T:B=S:J=1-2*(H=0):BEEP:GOTO 210
 280 GOSUB 330
 290 IF J<>0 THEN BEEP:GOTO 320
 300 CURSOR 2*X+2,3*Y+2:PEN 1-2*(H=0):IF H=1 TH
EN PRINT"*":ELSE PRINT"O"
 310 H=ABS(H-1):LASER:I=1
 320 GOTO 210
 330 J=POINT(16*X+25,210-24*Y)
 340 RETURN
```

Anagrammes

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexe (2 joueurs).
Difficulté du jeu : selon le niveau.

LE JEU :
Des lettres sont choisies au hasard par HECTOR, le nombre de lettres étant fonction du niveau de difficulté désiré. Puis les lettres du mot ainsi obtenu sont brouillées, et affichées au bas de l'écran ; un nouveau mot est alors confectionné, de façon aléatoire, à partir des mêmes lettres et il est affiché sous le premier, et ainsi de suite ; les deux joueurs (gauche et droit) disposent chacun d'une touche (1 ou Ø) et doivent la presser dès qu'ils voient s'afficher au bas de l'écran un mot déjà présent sur l'écran ; leur score est alors évalué, et un message correspondant à leur résultat sera délivré.

LE PROGRAMME :
Le niveau du jeu est entré à la ligne 5Ø. La boucle I (8Ø-1ØØ) réalise le choix des lettres.
Une deuxième boucle I (12Ø-17Ø) effectue le brouillage, et la ligne 14Ø permet de détecter la pression d'une touche.
La ligne 2ØØ affiche un nouveau mot, et fait monter d'une ligne sur l'écran les mots déjà obtenus.
La ligne 24Ø cherche quel est celui des deux joueurs qui a pressé une touche.
La boucle I (25Ø-27Ø) explore les mots présents sur l'écran ligne par ligne grâce au tableau A$. On obtient ainsi un nouveau mot qui est comparé au mot initial en ligne 26Ø et le résultat BRAVO apparaît si c'est exact.
Sinon, la partie s'arrête sur l'affichage PERDU (ligne 28Ø).

POUR JOUER :
On lance le programme par RUN.
On devra ensuite entrer le niveau de difficulté (en principe de 1 à 3) et faire ENTREE. Les mots s'affichent en partant du bas de l'écran, et montent au fur et à mesure ; si l'un des joueurs croit reconnaître dans le mot qui apparaît un des mots déjà affichés, il devra immédiatement presser sa touche (1 ou Ø), et son résultat sera affiché après un court instant.
La partie sera alors terminée.

EXTENSIONS POSSIBLES :
On pourra utiliser d'autres affichages que des mots, modifier les temps d'affichage, faire participer plus de deux joueurs, améliorer le son et la couleur, etc.

```
 10 'ANAGRAMMES
 20 DIM A$(20):SEEDTIME(0)
 30 STANDARD:PEN2:PAPERO
 40 CLS
 50 INPUT"NIVEAU (1 A 3) :";N
 60 IF N<1 OR N>3 THEN GOTO 40
 70 S$=""
 80 FOR I=1 TO N+3
 90 S$=S$+CHR$(INT(RND(65,91)))
 100 NEXT I
 110 A$="":B$=S$
 120 FOR I=1 TO LEN(S$)
 130 N=INT(RND(1,LEN(S$)+1))
 140 IF MID$(B$,N,1)="-" THEN GOTO 130
 150 A$=A$+MID$(B$,N,1)
 160 B$=LEFT$(B$,N-1)+"-"+MID$(B$,N+1)
 170 NEXT I
 180 D$=INKEY$(40)
 190 IF D$="1" OR D$="0" THEN GOTO 240
 200 BEEP:CURSOR 15,16:PRINTA$
 210 FOR I=1 TO 19:A$(I)=A$(I+1):NEXT I:A$(20)=
A$
 220 S$=A$
 230 GOTO 110
 240 IF D$="1" THEN H$="JOUEUR DE GAUCHE":ELSE
H$="JOUEUR DE DROITE"
 250 FOR I=1 TO 19
 260 IF S$=A$(I) THEN SHOT1:PEN 1:CURSOR N*6+40
,226-I*10:PRINT"BRAVO ";H$:GOTO 290
 270 NEXT I
 280 SHOTO:PEN 1:CURSOR N*6+40,16:PRINT"PERDU "
;H$
 290 PEN2
```

Œufs

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

LE JEU :
Un oiseau mystérieux se déplace en battant des ailes au hasard sur l'écran. Il pond des œufs (représentés par des *) que vous (un carré de couleur) devez essayer de récupérer. Votre score (nombre d'œufs récupérés) sera affiché en permanence, et la partie se déroule en un laps de temps défini.

LE PROGRAMME :
L'oiseau est dessiné en deux parties (pour le mouvement) aux lignes 7Ø et 11Ø ; son déplacement aléatoire est géré par le sous-programme 21Ø-3ØØ (en particulier les lignes 26Ø à 29Ø).
La ligne 13Ø permet de reboucler le programme sur la ligne 7Ø.
Le déplacement du joueur est régi par les lignes 15Ø et 16Ø, et le test permettant de détecter la présence d'un œuf est effectué en ligne 17Ø.
La ligne 19Ø permet l'affichage du score.
La fin de jeu est détectée en ligne 23Ø.

POUR JOUER :
On lancera le programme par RUN.
L'oiseau commence alors à se déplacer au hasard sur l'écran, et se met à pondre ses œufs. Lorsque le symbole qui représente le joueur (un carré de couleur) apparaît sur l'écran, celui-ci peut se déplacer en utilisant les 4 touches de déplacement. Il vous faut dès lors essayer de récupérer un maximum d'œufs avant la fin de la partie.
On pourra arrêter de jouer à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
Sur le même principe, on pourra facilement imaginer d'autres thèmes de jeu. On pourra bien sûr améliorer la couleur, les sons, le graphisme, et éventuellement, définir un jeu à plusieurs joueurs, chacun essayant de récupérer un maximum d'œufs.

```
 10 'OEUFS
 20 SEEDTIME(0)
 30 STANDARD:PEN2:PAPER0:COVER 1
 40 O$="*":O1$="'-'":O2$=".-."
 50 CLS:P=200:R=18:S=10:T=0:SOUND2,14
 60 X=INT(RND(0,34)):Y=INT(RND(0,19))
 70 CURSOR X*6+9,226-Y*10:PEN 1:PRINTO1$
 80 A$=INKEY$(40)
 90 IF ASC(A$)<>0 THEN GOSUB 140
 100 GOSUB 210
 110 CURSOR X*6+9,226-Y*10:PEN 1:PRINTO2$
 120 GOSUB210
 130 GOTO 70
 140 CURSOR R*6+9,226-S*10:PRINT" "
 150 R=R+(A$=CHR$(9))*(R<34)-(A$=CHR$(8))*(R>0)

 160 S=S+(A$=CHR$(30))*(S<19)-(A$=CHR$(29))*(S>
0)
 170 G=POINT(R*6+11,221-S*10)
 180 IF G=3 THEN T=T+1:BEEP
 190 CURSOR 9,11:PRINT"SCORE ... :";T;
 200 RETURN
 210 A=X:B=Y
 220 P=P-1
 230 IF P=0 THEN HUSH:PRINT:PRINT"C'EST FINI ..
.":LASER:LASER:COVER 0:END
 240 CURSOR R*6+9,226-S*10:PEN 2:PRINTCHR$(127)

 250 CURSOR A*6+9,226-B*10:PEN 3:PRINT" "+O$+"
"
 260 X=ABS(X+INT(1+2*(RND(0,1)<.5)))
 270 Y=ABS(Y+INT(1+2*(RND(0,1)<.5)))
 280 IF X>34 THEN X=1
 290 IF Y>19 THEN Y=1
 300 RETURN
```

Taquin de lettres

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :
Qui d'entre nous n'a pas, un jour ou l'autre, eu dans les mains un de ces petits puzzles où l'on essaie de déplacer d'une case à une autre des petits carrés de plastique colorés, jusqu'à l'obtention d'un résultat bien précis ? En son temps le 14-15 de SAM LOYD, a eu autant de succès que notre célèbre RUBIK'S.
Le taquin qui vous est proposé ici utilise 35 lettres, disposées au hasard dans un carré de 6 cases sur 6 cases, et ces lettres peuvent, replacées dans leur ordre initial, vous permettre de reconstituer un proverbe très connu. Une case coloriée vous permet de déplacer les lettres, et un compteur affiche en permanence le nombre d'essais : arriverez-vous à retrouver le proverbe ?

LE PROGRAMME :
La phrase à reconstituer (qui pourra bien sûr être modifiée) est placée dans une variable R$ en ligne 6Ø.
Le mélange des lettres et l'impression sont faits par la boucle I (7Ø-16Ø).
La ligne 19Ø réalise l'affichage du carré coloré ; celui-ci est déplacé par les lignes 23Ø et 24Ø en utilisant les 4 flèches.
Les lignes 25Ø et 26Ø réimpriment la lettre concernée à l'emplacement du carré coloré.
La ligne 28Ø comptabilise le nombre d'essais qui est imprimé grâce à la ligne 18Ø.

POUR JOUER :
On lance le programme par RUN.
La partie sera évidemment d'autant plus difficile que l'on ne connaît pas le proverbe à reconstituer. On peut déplacer les lettres et les faire passer à l'emplacement du carré coloré, à l'aide des 4 touches de direction (→, ←, ↓, ↑).
Le nombre d'essais est affiché en permanence, à droite du puzzle.
On peut arrêter à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
Le même principe peut être appliqué à toute sorte de taquins (manipulation de nombres, de couleurs, de sons éventuellement). On peut aussi prévoir plusieurs phrases, avec un choix effectué par un générateur aléatoire, etc.

```
 10 'TAQUIN DE LETTRES
 20 SEEDTIME(0)
 30 SPECIAL:DIM T$(5,5)
 40 WIPE:INK1:BOX 8,228,72,164:BOX88,228,216,10
0
 50 PAPER3:PENO
 60 A$="":R$="ONASOUVENTBESOIND'UNPLUSPETITQUES
OI"
 70 FOR I=1 TO 35
 80 X=INT(RND(1,36))
 90 IF MID$(R$,X,1)="-" THEN GOTO 80
 100 A$=A$+MID$(R$,X,1)
 110 R$=LEFT$(R$,X-1)+"-"+MID$(R$,X+1)
 120 F=ASC(MID$(A$,I,1))
 130 X=I-6*INT(I/6):Y=INT(I/6)
 140 SCALE1,1:CURSOR X+1,Y+1:PRINTCHR$(F)
 150 SCALE2,2:CURSOR X+6,Y+1:PRINTCHR$(F)
 160 T$(X,Y)=CHR$(F):NEXT I
 170 X=0:Y=0:S=0:C=32
 180 SCALE1,1:CURSORO,25:PAPERO:PEN2:PRINT"NOMB
RE D'ESSAIS :";S;
 190 CURSOR X+1,Y+1:PAPER3:PRINT" ":SCALE2,2:CU
RSORX+6,Y+1:PRINT" "
 200 T$=INSTR$(1):BEEP
 220 I=X:J=Y
 230 X=X+(T$=CHR$(9))*(X<5)-(T$=CHR$(8))*(X>0)
 240 Y=Y+(T$=CHR$(30))*(Y<5)-(T$=CHR$(29))*(Y>0
)
 250 C$=T$(X,Y):PEN 0:T$(X,Y)=T$(I,J):T$(I,J)=C
$
 260 SCALE1,1:CURSOR I+1,J+1:PRINTC$:SCALE2,2:C
URSOR I+6,J+1:PRINTC$
 270 IF I=X AND J=Y THEN SHOT1:GOTO 290
 280 S=S+1
 290 GOTO 180
```

Cibles

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : hasard.
Difficulté du jeu : facile.

LE JEU :
Il consiste simplement à tirer sur des cibles, en essayant de marquer le maximum de points ; le nombre de cibles (de 1 à 6) sera choisi en début de partie, et le score réalisé sera affiché en permanence. Si l'on touche le bord d'une cible, la partie s'arrête. On marquera d'autant plus de points que le nombre sur lequel on tombe sera plus grand.

LE PROGRAMME :
Le nombre de cibles est entré en ligne 5Ø. La boucle I (8Ø-1ØØ) effectue l'impression des cibles.
La ligne 12Ø affichera le score en permanence.
La balle est affichée par la ligne 13Ø.
La boucle I (15Ø-2ØØ) réalise l'impression des chiffres contenus dans les cibles, et permet le déplacement de la boule (ligne 19Ø).
La ligne 22Ø regarde si l'on a tiré, et la ligne 23Ø regarde si la boule est encore sur la ligne de départ.
La ligne 24Ø fait progresser la boule d'une case vers la droite, et la ligne 25Ø vérifie si l'on a ou non atteint les cibles ; puis les lignes 26Ø et 27Ø regardent ce que l'on a touché.
La ligne 28Ø incrémente le score.
Les lignes 29Ø et 3ØØ s'occupent de la fin de partie.

POUR JOUER :
Lancer le programme par RUN, et taper le nombre de cibles désiré suivi de ENTREE. Les cibles s'affichent, et la partie commence. Votre balle, représentée par un * est affichée en bas à gauche ; vous pouvez la déplacer avec les flèches haut et bas (touches ↑ et ↓), et vous pouvez tirer à tout moment en tapant Ø.
Le score sera modifié en fonction de votre résultat, et une nouvelle balle apparaît. Il est possible d'arrêter à tout moment en tapant CNT et C, sinon le jeu s'arrête au bout d'un certain nombre de tirs.

EXTENSIONS POSSIBLES :
On pourra sans difficulté augmenter le nombre de cibles, modifier la règle d'apparition des nombres dans les cibles pour la rendre moins hasardeuse, s'arranger pour que la balle elle-même soit représentée par un chiffre qui devrait coïncider à l'arrivée avec celui de la cible.

```
10 'CIBLES
20 SEEDTIME(0)
30 SPECIAL:SCALE1,1
40 CLS
50 PRINT:INPUT"--- NB DE CIBLES :";N
60 P=10:J=26:S=0:B=1
70 CLS:PAPER3:PEN1
80 FOR I=1 TO N
90 CURSOR J,3*I-1:PRINT"=:":CURSOR J,3*I:PRINT
" :":CURSOR J,3*I+1:PRINT"=:"
100 NEXT I
110 K=0:A=18:PEN2:PAPER0
120 CURSOR 1,25:PRINT"SCORE :";S;"  BALLE #";B

130 CURSOR K,A:PEN 1:PRINT"*"
140 D=A
150 FOR I=1 TO N
160 U(I)=INT(RND(0,P))
170 CURSOR J,3*I:PRINTCHR$(U(I)+48)
180 A$=INKEY$(30)
190 A=A-(A$=CHR$(30))+(A$=CHR$(29))
200 NEXT I
210 CURSOR K,D:PRINT" "
220 IF A$="0" THEN B=B+1:GOTO 240
230 IF K=0 THEN GOTO 130
240 K=K+1
250 IF K<J THEN GOTO 130
260 FOR I=1 TO N:IF A=I*3 THEN C=I:I=N:NEXT I:
GOTO 280
270 NEXT I:PRINT"PERDU ...":STANDARD:END
280 S=S+U(C)
290 P=P-1
300 IF P=0 THEN SHOT1:END
310 GOTO 110
```

Gouttes

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : dessin.
Difficulté du jeu : aucune.

LE JEU :
C'est un véritable programme de dessin animé que l'on peut considérer, en fait, comme un interlude ; un récipient est dessiné sur l'écran, vu en coupe, et il contient déjà des obstacles. Une goutte d'eau, symbolisée par une étoile jaune ou rouge, apparaît au-dessus du récipient et tombe en remplissant le récipient au fur et à mesure. Sa trajectoire, lorsqu'elle rencontre un obstacle, est entièrement déterminée par le hasard : elle part à droite ou à gauche, jusqu'à ce qu'il n'y ait plus aucune possibilité. Alors, une autre goutte tombe, et ainsi de suite, jusqu'au débordement final.

LE PROGRAMME :
Les deux boucles emboîtées I (60-120) et J (70-110) dessinent le récipient et les obstacles.
La ligne 170 dessine la goutte et la ligne 210 envoie dans un sous-programme (370-390) qui permet de déterminer s'il y a ou non rencontre avec un obstacle.
Si c'est non, on va en 290, et la goutte continue de tomber.
Si c'est oui, la goutte part à droite ou à gauche (ligne 230) selon la valeur définie pour K en ligne 150.
On regarde de nouveau s'il y a ou non obstacle, ou si la goutte peut continuer à tomber (lignes 250 ou 260).
Selon le résultat du test, on utilisera 270, 290, 310 ou 340.

POUR JOUER :
Le RUN déclenche le dessin du récipient et des obstacles, la goutte apparaît, se met à tomber et finit par se trouver une place. Une autre goutte apparaît et l'on recommence.
Le programme s'arrête de lui-même lorsque le récipient déborde ; on peut l'arrêter à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
On peut, bien entendu, ajouter la couleur et le son et aussi envisager un jeu dans lequel un petit personnage devrait échapper aux gouttes ou à la noyade.

```
 10 'GOUTTES
 20 SEEDTIME(0)
 30 SPECIAL:SCALE1,1
 40 X$="*"
 50 CLS
 60 FOR I=12 TO 20
 70 FOR J=6 TO 21
 80 PAPER 2:CURSOR 5,I:PRINT" ":CURSOR 22,I:PRI
NT" "
 90 CURSOR J,20:PRINT" "
 100 IF I<>20 AND I<>12 AND RND(0,1)<.3 THEN CU
RSOR INT(RND(6,22)),I:PRINT" "
 110 NEXT J
 120 NEXT I
 140 Y=INT(RND(10,18)):X=6
 150 K=-2*(RND(0,1)<.5)-1
 160 I=0
 170 CURSOR Y,X:PAPER 0:PEN INT(RND(0,2))*2+1:P
RINTX$
 175 BEEP
 190 CURSOR Y,X:PRINT" "
 200 U=X+1:V=Y
 210 GOSUB 370
 220 IF J=0 THEN GOTO290
 230 U=X:V=Y+K
 240 GOSUB 370
 250 IF J=0 THEN GOTO 310
 260 IF I=0 THEN GOTO 340
 270 CURSOR Y,X:PAPER 0:PEN INT(RND(0,2))*2+1:P
RINTX$
 280 GOTO 140
 290 X=X+1
 300 GOTO 150
 310 Y=Y+K
 320 I=1
 330 GOTO 170
 340 K=-K
 350 I=1
 360 GOTO 230
 370 P=POINT(V*8+11,224-U*8)
 380 J=-(P<>0):IF U=24 THEN END
 390 RETURN
```

Obstacles

REFERENCES :
Difficulté de programmation : assez difficile.
Catégorie de jeu : jeu de réflexe (pour un joueur).
Difficulté du jeu : moyenne.

LE JEU :
Des piétons (*) se déplacent sur l'écran pour venir à la rencontre d'une petite voiture (#) qui dispose de la possibilité de se déplacer vers la gauche ou vers la droite pour les éviter et qui devra tenir le plus longtemps possible.
La distance parcourue sera affichée en permanence, et le jeu se termine lorsque la voiture rencontre, soit le bord du terrain, soit un piéton.

LE PROGRAMME :
La ligne 8Ø fait avancer les piétons vers la voiture du joueur qui est imprimée grâce à la ligne 7Ø.
Les lignes 8Ø, 9Ø, 1ØØ fabriquent la route (les lignes 9Ø et 1ØØ bien qu'identiques sont toutes deux indispensables, car la valeur de RND ne sera, en principe, pas la même sur ces deux lignes).
La ligne 12Ø affiche le score.
La ligne 14Ø permettra de déplacer le véhicule.
Les lignes 15Ø et 16Ø permettent de contrôler s'il y a ou non un obstacle et arrêtent le programme s'il y a lieu.
Le score est incrémenté en ligne 17Ø.

POUR JOUER :
On lance le programme en tapant RUN.
La route et les piétons apparaissent en bas de l'écran et montent vers la voiture. La voiture peut se déplacer à l'aide des flèches de déplacement horizontal (touches → et ←).
La rencontre avec un piéton ou la bordure interrompt le déroulement du programme et provoque l'affichage du score.
On peut, cependant, sortir du programme à tout moment en utilisant les touches CNT et C.

EXTENSIONS POSSIBLES :
Il sera assez facile d'améliorer le dessin de la voiture, de la route, des piétons. On pourra améliorer le son et la couleur, admettre un certain nombre d'accidents, enregistrer le meilleur score (pour plusieurs joueurs), éventuellement dessiner plusieurs voitures, etc.

```
 10 'OBSTACLES
 20 SEEDTIME(0)
 30 STANDARD:PEN2:PAPER0
 40 V$="#":P$="*"
 50 CLS
 60 S=0:Y=5
 70 CURSOR Y*6+9,216:PEN3:PRINTV$
 80 CURSOR 9,16:PEN 1:PRINT "          ";
 90 IF RND(0,1)<.7 THEN CURSOR INT(RND(1,9))*6+
9,16:PRINTP$;:BEEP
 100 IF RND(0,1)<.8 THEN CURSOR INT(RND(1,9))*6
+9,16:PRINTP$:BEEP
 120 CURSOR 100,226:PRINT"SCORE :";S
 130 A$=INKEY$(15)
 140 Y=Y+(A$=CHR$(9))*(Y<9)-(A$=CHR$(8))*(Y>0)
 150 C=POINT(6*Y+11,204)
 160 IF C=1 THEN SHOT1:PEN2:END
 170 S=S+1
 180 GOTO 70
```

Captures dans l'espace

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexe (1 seul joueur).
Difficulté du jeu : moyenne.

LE JEU :
Votre vaisseau spatial navigue dans un dangereux champ de météorites et votre mission est de vous emparer de la plus grande quantité possible de minerai d'urane, un métal indispensable que l'on ne peut trouver que dans ces lieux particulièrement dangereux.
Vous devez arriver à manœuvrer pour recueillir l'urane, sans vous faire heurter par une météorite.

LE PROGRAMME :
La ligne 7Ø permet de faire apparaître les météorites.
La ligne 9Ø fera apparaître le minerai d'urane et assure déplacement des météorites sur l'écran.
Les lignes 1ØØ à 12Ø contrôlent le déroulement du jeu (a-t-on heurté une météorite ou a-t-on capturé un bloc d'urane ?).
La rencontre d'une météorite provoque la fin de partie.
Votre vaisseau spatial est imprimé par les lignes 13Ø et 14Ø et son déplacement est contrôlé par la ligne 16Ø.

POUR JOUER :
On lance le programme par RUN et les météorites commencent à monter sur l'écran, ainsi que les blocs d'urane.
Les touches de déplacement horizontal (→ et ←) seront utilisées pour déplacer votre vaisseau et récupérer l'urane, mais le moindre heurt avec une météorite provoquera la fin de la partie et l'affichage de la quantité d'urane capturée.
On pourra aussi arrêter à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
On peut rendre le jeu de plus en plus difficile en augmentant progressivement le nombre de météorites, ou encore, en minutant la durée de la partie.
On peut aussi prévoir un affichage permanent du score, introduire un deuxième vaisseau spatial manœuvré par une puissance étrangère et qui cherche aussi à s'emparer de l'urane...

```
10 'CAPTURES DANS L'ESPACE
20 SEEDTIME(0)
30 STANDARD:PEN2:PAPER0
40 CLS
50 X=15:Z=0
60 S=INT(RND(0,38))
70 BEEP:CURSOR S*6+9,16:PEN 1:PRINT"#";
80 S=INT(RND(0,38))
90 IF RND(0,1)<.7 THEN BEEP:CURSOR S*6+9,16:PE
N 3:PRINT"*"
100 C=POINT(X*6+11,104)
110 IF C=1 THEN PEN2:PRINT:PRINT"SCORE OBTENU
:";Z:END
120 IF C=3 THEN LASER:Z=Z+1
130 CURSOR X*6+3,116:PRINT"    "
140 PEN 2:CURSOR X*6+3,106:PRINT" "
150 A$=INKEY$(15)
160 X=X+(A$=CHR$(9))*(X<36)-(A$=CHR$(8))*(X>0)

170 GOTO 60
```

Anomalie

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

LE JEU :
Il repose essentiellement sur une bonne capacité d'observation. Deux grilles de 25 cases chacune sont dessinées sur l'écran. Des symboles identiques sont affichés en des endroits différents des deux grilles, mais l'un des symboles a été modifié sur la grille de droite. Le jeu consiste à trouver lequel, en entrant ses coordonnées dans l'ordre ligne-colonne.

LE PROGRAMME :
La boucle I (5Ø-8Ø) imprime les chiffres de repérage des grilles.
La boucle principale I (1ØØ-16Ø) imprime les symboles des deux grilles, dans un ordre différent.
La ligne 11Ø choisit le caractère à imprimer.
Le sous-programme (1ØØØ-1Ø5Ø) s'occupe de faire l'impression à un emplacement libre (on l'appelle deux fois pour les deux grilles).
Les lignes 17Ø et 18Ø changent, de façon aléatoire, l'un des caractères.
Les coordonnées du point essayé sont entrées aux lignes 19Ø et 2ØØ, et le contrôle est effectué par la ligne 21Ø.
Les lignes 22Ø et 23Ø s'occupent de l'affichage du résultat.

POUR JOUER :
Après avoir lancé le programme par l'instruction RUN, on obtient l'affichage des deux grilles. On devra alors chercher, par comparaison, quel est le caractère qui a été modifié (sur la grille de droite).
On entre alors son numéro de ligne, suivi de ENTREE, et son numéro de colonne, suivi de ENTREE.
Le programme répondra alors BRAVO ou PERDU.

EXTENSIONS POSSIBLES :
Il sera possible d'utiliser des symboles graphiques de diverses couleurs, ce qui pourra simplifier la recherche.
On peut aussi envisager d'utiliser des dessins plus sophistiqués, améliorer le contrôle du résultat, admettre la bonne réponse sur la grille de droite ou celle de gauche, introduire un minutage, etc.

```
 10 'ANOMALIE
 20 SEEDTIME(0):DIM T(5,5)
 30 STANDARD:PEN2:PAPER0:COVER1
 40 CLS:T=5
 50 FOR I=1 TO 5
 60 CURSOR I*12+15,226:PRINTI:CURSOR 9,226-20*I
:PRINTI
 70 CURSOR I*12+123,226:PRINTI:CURSOR 115,226-2
0*I:PRINTI
 80 NEXT I
 90 PEN 1
 100 FOR I=1 TO 22
 110 D=INT(RND(1,27))
 120 L=2:S=T
 130 GOSUB1000
 140 L=11
 150 GOSUB 1000
 160 NEXT I
 170 S=0
 180 GOSUB 1000
 190 PEN2:PAPER 0:CURSOR 9,66:INPUT"LIGNE ... :
";I
 200 CURSOR 9,50:INPUT"COLONNE . :";T
 210 C=T(T,I)
 220 IF D=C THEN PRINT"BRAVO ...":ELSE PRINT"PE
RDU ..."
 230 PRINT:PRINT"LA LETTRE A TROUVER ETAIT ";CH
R$(D+64)
 240 COVER 0
 250 END
 1000 U=2*INT(RND(1,T+1)):V=2*INT(RND(L,L+S))
 1010 C=POINT(V*6+11,224-U*10)
 1020 IF C<>0 AND S<>0 THEN GOTO 1000
 1030 PAPER3:PEN1:CURSOR V*6+9,226-U*10:PRINTCH
R$(D+64)
 1040 IF L=11 THEN T((V-19)/2,U/2)=D
 1050 RETURN
```

Plusieurs en un

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : deux joueurs.
Difficulté du jeu : variable, selon l'utilisation.

LE JEU :
Plusieurs jeux sont en réalité possibles avec ce programme qui va se comporter comme un crayon électronique vous permettant d'inscrire des symboles O ou X, en n'importe quel emplacement de votre feuille-écran de télévision. On pourra, par exemple, l'utiliser pour un jeu de morpion (le premier à réaliser un alignement de 5 pions identiques gagne) ou pour un GO (on réalise des encadrements des pions adverses) ou tout autre jeu, selon son imagination.

LE PROGRAMME :
Le changement de joueur est effectué par la ligne 5Ø.
Le joueur qui a le trait voit le symbole qu'il a choisi (O ou X) se placer en limite du cadre (ligne 7Ø).
Les lignes 1ØØ et 11Ø permettent aux deux joueurs de déplacer le symbole sur le terrain.
Les lignes 13Ø et 14Ø contrôlent si le déplacement est possible (case libre et non dépassement des limites).
La ligne 19Ø permet d'écrire sur une case libre.

POUR JOUER :
RUN provoque l'affichage des limites du terrain et du numéro du joueur, ainsi que des symboles correspondants (O ou X) que l'on devra déplacer sur l'écran à l'aide des 4 touches de direction (→, ←, ↓, ↑).
Il n'est pas possible de venir sur une case déjà occupée et il faudra appuyer sur la touche Ø pour inscrire son propre symbole dans une case disponible.
Dès que cela a été fait, le symbole du second joueur apparaît dans la colonne de gauche, à peu près au milieu, et le joueur procédera de la même manière que son adversaire.
Pour sortir du programme, il suffira de taper CNT et C.

EXTENSIONS POSSIBLES :
On pourra utiliser l'écran tout entier comme terrain de jeu et réaliser des contrôles en fonction des différents jeux que l'on désire pratiquer : contrôle d'alignements de 5 pièces pour le morpion, contrôle du nombre de pièces capturées pour le GO.
On pourra utiliser le crayon optique pour désigner la case à occuper.

```
 10 'PLUSIEURS EN UN
 20 STANDARD:PEN2:PAPERO:CLS:COVER 1
 30 C$(0)="*":C$(1)="O":C(0)=1:C(1)=3
 40 J=0
 50 J=ABS(J-1)
 60 X=12:Y=0
 70 CURSOR Y*6+9,226-X*10:PAPER C(J):PEN 0:PRIN
TC$(J)
 80 A$=INSTR$(1)
 90 A=X:B=Y
 100 X=X+(A$=CHR$(30))*(X<19)-(A$=CHR$(29))*(X>
0)
 110 Y=Y+(A$=CHR$(9))*(Y<36)-(A$=CHR$(8))*(Y>0)

 120 PAPERO:CURSOR B*6+9,226-A*10:PRINT" "
 130 C=POINT(Y*6+9,226-X*10)
 140 IF C<>1 AND C<>3 THEN GOTO 170
 150 X=A:Y=B
 170 IF A$<>"0" OR Y=0 THEN GOTO 70
 180 CURSOR Y*6+9,226-X*10
 190 PAPER C(J):PRINTC$(J)
 200 IF J=0 THEN BEEP:ELSE BELL
 210 GOTO 50
```

Parcours du cavalier

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :
Il se présente, ici, comme un jeu de hasard total, mais provient, en réalité, d'un problème assez difficile à résoudre : comment déplacer un cavalier sur un échiquier de façon qu'il passe une fois, et une seule, sur chacune des 64 cases du damier.
Le cavalier se déplace suivant les règles communément admises aux échecs, et sa position de départ est définie par le joueur ; par contre, ses déplacements sont effectués au hasard. Arrivera-t-il à parcourir les 64 cases ?

LE PROGRAMME :
La ligne 6Ø définit les divers déplacements possibles, dans deux variables alphanumériques A$ et B$; la boucle I (7Ø-1ØØ) dessine le terrain de jeu.
On entre les coordonnées de la case de départ (ligne, puis colonne) aux lignes 11Ø et 12Ø.
La ligne 14Ø imprime le numéro du coup.
La ligne 15Ø permet le tirage au hasard d'un déplacement du cavalier.
La boucle Y (16Ø-21Ø) cherche si le déplacement est possible ; si ce n'est pas le cas, on essaie tous les autres (toujours dans le même ordre).
Les lignes 17Ø et 18Ø effectuent le déplacement.
La ligne 19Ø contrôle que l'on ne sorte pas du terrain de jeu.
La ligne 2ØØ envoie en ligne 23Ø si le coup est possible.
La ligne 22Ø gère la fin de partie (cavalier bloqué) et affiche le score.
Les lignes 23Ø, 24Ø et 25Ø réalisent le coup et incrémentent le score.

POUR JOUER :
On lance le programme par l'ordre RUN. Le terrain de jeu s'affiche ; on choisit alors la position de départ en tapant d'abord un numéro de ligne (entre 1 et 8), suivi de ENTREE puis son numéro de colonne, suivi de ENTREE. Le jeu commence immédiatement et il se déroule de façon entièrement automatique jusqu'à ce que le cavalier soit bloqué. On peut, bien sûr, arrêter à tout moment par CNT et C.

EXTENSIONS POSSIBLES :
On peut imaginer de nombreuses variantes sur ce thème, ne serait-ce qu'en commençant par faire varier la taille des échiquiers, on peut aussi améliorer considérablement le graphisme. Mais, le plus intéressant est d'arriver à définir un algorithme donnant la ou les solutions du problème initial...

```
 10 'PARCOURS DU CAVALIER
 20 STANDARD:PEN2:PAPERO:CLS:COVER 1
 30 SEEDTIME(0)
 40 CLS
 50 S=1:T=0
 60 A$="56653223":B$="653223"+A$+"56":A$=A$+A$
 70 FOR I=1 TO 8
 80 CURSOR I*18+15,226:PRINTI
 90 CURSOR 9,226-I*20:PRINTI;"  .  .  .  .  .
.  .  ."
 100 NEXT I
 110 PAPERO:CURSOR 9,46:INPUT"LIGNE DE DEPART .
.... :";I
 120 CURSOR 9,26:INPUT"COLONNE DE DEPART ... :"
;J
 130 IF S>9 THEN S=0:T=T+1
 140 BEEP:PAPER1:CURSOR J*18+21,226-I*20:PRINTR
IGHT$(STR$(S),1):IF T<>0 THEN CURSOR J*18+15,22
6-I*20:PRINTRIGHT$(STR$(T),1)
 150 Z=INT(RND(1,9))
 160 FOR Y=Z TO Z+8
 170 A=I+VAL(MID$(A$,Y,1))-4
 180 B=J+VAL(MID$(B$,Y,1))-4
 190 IF A<1 OR B<1 OR A>8 OR B>8 THEN GOTO 210
 200 IF POINT(B*18+21,226-A*20)=0 THEN Y=Z+8:NE
XT Y:GOTO 230
 210 NEXT Y
 220 CURSOR 160,46:PRINT"TOTAL :";T*10+S:PAPERO
:END
 230 I=A:J=B
 240 S=S+1
 250 GOTO 130
```

Mur de briques

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexe (pour 1 joueur).
Difficulté du jeu : difficile.

LE JEU :
S'il est un jeu que l'on peut présenter comme un grand classique des jeux d'arcades pour micro-ordinateur, c'est bien le mur de briques (en anglais BREAKOUT).
Il est offert, ici, dans une version simplifiée, mais offre néanmoins de bonnes performances : affichage du score en fin de partie, rebonds aléatoires, etc.
La règle en est très élémentaire : vous disposez d'une raquette qui peut être déplacée horizontalement pour renvoyer une balle (matérialisée par un point) qui va démolir les briques dessinées en haut de l'écran.
Votre problème est d'en démolir le plus grand nombre possible, sans rater la balle, ce qui provoquerait la fin de partie et l'affichage du score.

LE PROGRAMME :
La boucle I (5Ø-8Ø) dessine le terrain de jeu et les briques sont mises en place par les boucles emboîtées I (9Ø-13Ø) et J (1ØØ-12Ø).
La ligne 15Ø dessine la raquette dont la position est contrôlée par la ligne 17Ø.
Les lignes 22Ø à 25Ø regardent si l'on a touché le mur, les briques ou la raquette.
Les rebonds sont gérés en conséquence avec des éventuels aléas, par les lignes 26Ø à 3ØØ.
La ligne 19Ø affiche le score en permanence.

POUR JOUER :
Le programme sera lancé par l'instruction RUN, ce qui provoque l'affichage du terrain de jeu puis, très rapidement, l'envoi de la balle. Le joueur devra alors essayer de déplacer sa raquette à l'aide des touches de déplacement horizontal (→ et ←).
Si, à un moment quelconque, sa balle ne touche pas la raquette, le score sera affiché et la partie s'arrête.
On peut aussi terminer à tout moment par CNT et C.

EXTENSIONS POSSIBLES :
On peut envisager d'agrandir le terrain de jeu, d'augmenter le nombre de briques, de prévoir plusieurs balles, d'introduire des effets sonores.
On peut aussi, en s'inspirant des jeux d'arcades, introduire de petits ouvriers qui viendront reconstruire votre mur de briques, au fur et à mesure que vous le démolissez.

```
 10 'MUR DE BRIQUES
 20 STANDARD:PEN2:PAPER0:CLS:COVER 1
 30 SEEDTIME(0)
 40 CLS
 50 FOR I=1 TO 18
 60 CURSOR I*6+63,226:PRINT CHR$(127)+CHR$
(127)+CHR*(127);
 70 CURSOR 69,226-I*10:PRINT":CURSOR 171,226-
I*10:PRINT"
 80 NEXT I
 90 PEN 1:FOR I=1 TO 7
 100 FOR J=11 TO 26
 110 CURSOR J*6+9,226-I*10:PRINT"#"
 120 NEXT J
 130 NEXT I
 140 H=-1:K=-.5:J=18:I=17:Y=J
 150 CURSOR Y*6-3,36:PEN3:PRIN "
 160 A$=INKEY$(20)
 170 Y=Y+(A$=CHR$(9))*(Y<28)-(A$=CHR$(8))*(Y>8)

 180 IF T=1 THEN CURSOR INT(J)*6+9,226-I*10:PRI
NT" ";
 190 CURSOR 15,26:PRINT"SCORE :";S;
 200 IF I>21 THEN SHOT1:PEN2:END
 210 I=I+H:J=J+K
 220 C=POINT(INT(J)*6+11,224-I*10)
 230 IF C=0 THEN PEN2:CURSOR INT(J)*6+9,226-I*1
0:PRINT"O";:T=1:GOTO 150
 235 BEEP
 240 IF C=2 THEN GOTO 280
 250 IF C=1 THEN F=I:CURSOR INT(J)*6+9,226-I*10
:PEN 2:PRINT"O";:T=1:GOTO 300
 260 H=-H
 270 K=RND(0,1)-RND(0,1)
 280 K=-K:T=0
 285 PRINTK
 290 GOTO 150
 300 S=S+F
 310 K=-(K-(2*(RND(0,1)<.5)+1)*RND(0,1)/3)
 320 H=-H
 330 GOTO 150
```

Othello

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : difficile.

LE JEU :
Devenu un grand classique des jeux sur micro-ordinateur, le jeu d'Othello ou Reversi gagne, même si comme ici il est pratiqué à deux joueurs, à être géré sur ordinateur, avec visualisation : en effet, le jeu consistant essentiellement en des retournements des pions de l'adversaire, ceci peut se révéler, à la longue, très fastidieux, surtout si vous êtes un joueur assidu.
Le principe est que le joueur possédant les X doit pour pouvoir jouer, réussir à encadrer une ou plusieurs des pièces de son adversaire, dans une ou plusieurs directions ; s'il y réussit, les O seront alors retournés pour devenir des X (le retournement pourra être fait horizontalement, verticalement ou suivant les diagonales).
Le joueur ayant les O procédera de la même façon, et ceci jusqu'à ce que le damier soit rempli ou que la personne ne puisse plus jouer (si un joueur ne peut retourner des pions, il doit passer son tour).

LE PROGRAMME :
Les boucles emboîtées P (5Ø-12Ø) et Q (7Ø-11Ø) dessinent le terrain de jeu et la ligne 14Ø affiche un X ou un O, selon le joueur qui commence.
La boucle P (2ØØ-32Ø) contrôle la validité du coup (la case est-elle libre : lignes 3ØØ et 31Ø), en utilisant une partie de programme qui se trouve aux lignes 36Ø à 42Ø (a-t-on pris un ou plusieurs ennemis en ligne 36Ø et retournement éventuel autorisé par la ligne 38Ø : on refait un balayage dans toutes les directions, si l'encadrement s'est révélé possible).
La ligne 2ØØ définit les 8 directions d'exploration.
Les lignes 24Ø et 25Ø calculent les déplacements à effectuer pour explorer ces 8 directions.
La ligne 23Ø réalise les déplacements.
La ligne 34Ø assure le changement de joueur, s'il y a eu retournement (contrôlé en ligne 33Ø).

POUR JOUER :
Après RUN, chaque joueur entrera à son tour les coordonnées (ligne, ENTREE, puis colonne, ENTREE) de la case où il désire jouer ; si le coup est accepté, les pions sont retournés et l'on change de joueur ; si un coup n'est pas possible, on entre Ø en première coordonnée pour le changement de joueur.

EXTENSIONS POSSIBLES :
Avant tout, bien sûr, améliorer l'affichage avec des couleurs, une grille plus aérée.
On pourra aussi imaginer un jeu machine pour HECTOR.
Après avoir lancé le programme par RUN, il convient d'entrer la dimension de la grille (4, ENTREE ou 6, 8 ...).

```
 10 'OTHELLO
 20 STANDARD:PEN2:PAPER0:CLS:COVER 1
 30 INPUT"DIMENSION DE LA GRILLE :";A:IF A/2<>I
NT(A/2) THEN BEEP:GOTO 20
 40 CLS
 50 FOR P=1 TO A
 60 CURSOR P*12+9,226:PAPER0:PEN2:PRINTRIGHT$(S
TR$(P),1)
 70 FOR Q=1 TO A
 80 CURSOR 9,226-P*20:PAPER0:PEN2:PRINTRIGHT$(S
TR$(P),1):CURSOR Q*12+9,226-P*20:PRINT"."
 90 IF Q=A/2+1 AND P=A/2 THEN CURSOR Q*12-3,226
-P*20:PAPER1:PEN0:PRINT"O":PAPER3:CURSOR Q*12+9
,226-P*20:PRINT"*"
 100 IF Q=A/2+1 AND P=A/2+1 THEN CURSOR Q*12-3,
226-P*20:PAPER3:PEN0:PRINT"*":PAPER1:CURSOR Q*1
2+9,226-P*20:PRINT"O"
 110 NEXT Q
 120 NEXT P
 130 K=1
 140 CURSOR 130,206:PEN0:PAPER1-2*(K=1):IF K=1
THEN PRINT"JOUEUR *":ELSE PRINT"JOUEUR O"
 150 PAPER0:PEN2:CURSOR130,186:PRINT"
       ":CURSOR 130,166:PRINT"
"
 160 CURSOR130,186:INPUT"LIGNE :";U
 170 IF U=0 THEN GOTO 340
 180 CURSOR130,166:INPUT"COLONNE :";V
 190 H=1
 200 FOR P=1 TO 8
 210 S=0
 220 W=0
 230 T=2*U:R=2*V
 240 F=-(P>3)*(P<7)+  (P<3 OR P>7)/65535
 250 G=-(P>1)*(P<5)-(P>=6)
 260 Q=46-42*(S=1)-33*(S=2)
 270 IF S=0 THEN GOTO 290
 280 CURSOR R*6+9,226-T*10:PEN 0:PAPER 1-2*(K=1
):IF K=1 THEN PRINT"*":ELSEPRINT"O"
 290 T=T+2*F:R=R+2*G
 300 C=POINT(R*6+9,226-T*10)
 310 IF C=1 OR C=3 THEN GOTO 360
 320 NEXT P
 330 IF H<>0 THEN GOTO 140
 340 K=ABS(K-1)
 350 GOTO 140
 360 IF (C=1 AND K=1) OR (C=3 AND K=0) THEN W=1

 370 IF S<>0 THEN CURSOR R*6+9,226-T*10:PAPER 1
-2*(K=1):IF K=1 THEN PRINT"*":ELSE PRINT"O"
 380 IF W=1 AND ((C=1 AND K=1) OR (C=3 AND K=0)
) THEN GOTO 290
 390 IF W=0 OR S<>0 THEN GOTO 320
 400 S=1-(K=0)
 410 H=-H*(S=0)
 420 GOTO 220
```

Achevé d'imprimer
sur les presses de l'imprimerie IBP
à Rungis (Val-de-Marne 94) 686.73.54
Dépôt légal - mars 1985

N° d'impression: 4843
N° d'édition: 86595-237-1
N° d'ISBN: 2-86595-237-1

Votre avis nous intéresse

- Pour nous permettre de faire de meilleurs livres, adressez-nous vos critiques sur le présent livre.
- Si vous souhaitez des éclaircissements techniques, écrivez-nous, nous adresserons votre demande à l'auteur qui ne manquera pas de vous répondre directement.

- Ce livre vous donne-t-il toute satisfaction?

____

____

____

____

____

- Y a-t-il un aspect du problème que vous auriez aimé voir abordé?

____

____

____

____

____

Comment avez-vous eu connaissance de ce livre?

☐ publicité
☐ catalogue
☐ boutique micro
☐ autres
☐ cadeau
☐ librairie
☐ exposition

Avez-vous déjà acquis des livres PSI?

lesquels? ______________________________

____

qu'en pensez-vous? ______________________________

____

____

Nom ______________ Prénom ______________ Age ______
Adresse ______________________________
Profession ______________________________
Centre d'intérêt ______________________________

CATALOGUE GRATUIT

Vous pouvez obtenir un catalogue complet des ouvrages PSI, sur simple demande, ou en retournant cette page remplie à votre libraire, à votre boutique micro ou aux

Editions du PSI
BP 86
77402 Lagny-sur-Marne Cedex

Apprendre en se distrayant, tel est l'objectif de ce livre. 102 programmes de jeu vous guideront progressivement dans l'exploration du **BASIC 3X.**

Niveau par niveau, vous assimilerez de nouvelles instructions pour arriver à la maîtrise de votre **HECTOR HRX** ou **MX.** Cet ouvrage s'adresse aussi aux possesseurs du modèle 2HR+ qui n'auront aucune difficulté à passer du BASIC 3X au BASIC III.

Chaque niveau commence par une présentation pédagogique du jeu d'instructions utilisé. Tous les jeux sont décrits et les programmes analysés ligne par ligne afin d'en faciliter la modification.

HECTOR n'attend plus que vous pour jouer et programmer !

DES LIVRES POUR VOTRE ORDINATEUR

ISBN : 2.86595.237.1
120 FF

9 782865 952373

Couverture GIBRAT

102 PROGRAMMES POUR HECTOR EN BASIC 3X

PAR JACQUES DECONCHAT

PROGRAMMES

P.S.I.